The Acquisition of Valued Phenotypes

Dissertation

zur

Erlangung der Naturwissenschaftlichen Doktorwürde

(Dr. sc. nat.)

vorgelegt der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Jeremiah Wright

aus den

Vereinigten Staaten von Amerika

Promotionskomitee:

Prof. Dr. Andreas Wagner (Vorsitz)

Prof. Dr. Martin Ackermann

Prof. Dr. Jack Pronk

Zürich, 2012

Abstract

All organisms possess a set of physical traits, and the *phenotype* is the subset of those traits that are observable. The phenotypic traits of some organisms are valued by people, and Article 1 describes two biological techniques that can be used to acquire valued phenotypes – *artificial selection*, where the propagation of particular existing phenotypes is encouraged, and *induced phenotypic transition*, where new phenotypes are produced from existing phenotypes. Both of these techniques depend upon interactions between organisms and their environment, and to improve our understanding of those interactions, I developed a free, open-source software platform called the Systems Biology Research Tool, which is described in Article 2. The Systems Biology Research Tool primarily contains implementations of algorithms for simulating the behavior of chemical reaction networks, such as metabolic networks, and for identifying particular interaction patterns therein. Article 3 describes an algorithm for identifying a particular pattern – loops, or structural cycles – in large chemical reaction networks. Knowledge of these cycles can be used to improve the predictive ability of phenotypic simulations and to identify potential feedback loops in cellular processes. In Article 4, two laboratory techniques for the acquisition of a valued microbial phenotype are described. Both techniques utilize artificial selection and depend upon phenotypic transitions, and one involves a feedback loop between a microbial population's metabolic phenotype and its chemical environment.

Abstrakt

Alle Organismen besitzen eine Reihe von Merkmalen und die sichtbare Teilmenge dieser Merkmale ist der Phänotyp. Der Phänotyp einiger Organismen werden von Personen bewertet. Kapitel 1 beschreibt zwei Techniken mit Hilfe derer diese bewertbaren Phänotypen erlangt werden können. Eine dieser Techniken ist die künstliche Selektion. Hierbei wird die Ausbreitung von bereits vorhandenen Phänotypen gefördert. Die andere Technik ist der induzierte phänotypische Übergang. Hierbei werden neue Phänotypen aus bereits vorhandenen Phänotypen erzeugt. Beide Techniken sind von den Wechselwirkungen zwischen Organismen und ihrer Umwelt abhängig. Um unser Verständnis dieser Wechselwirkungen zu verbessern, habe ich eine kostenlose Open-Source-Software-Plattform namens "Systems Biology Research Tool" entwickelt, welches in Kapitel 2 beschrieben ist. Das "Systems Biology Research Tool" enthält in erster Linie implementierte Algorithmen zur Simulation des Verhaltens von chemischen Reaktionsnetzwerken, wie z.B. metabolischen Netzwerken, und zur Identifizierung bestimmter Interaktionsmuster in diesen Netzwerken. Kapitel 3 beschreibt einen Algorithmus zur Identifizierung bestimmter Muster in großen chemischen Reaktionsnetzwerken, wie z.B. Schleifen oder strukturelle Zyklen. Das Wissen um diese Zyklen kann benutzt werden um die voraussagende Fähigkeiten von phänotypischen Simulationen zu verbessern und potentielle Feedback-Schleifen in zellulären Prozessen zu identifizieren. In Kapitel 4 werden zwei Labortechniken für den Erwerb eines bewertbaren mikrobiellen Phänotyp beschrieben. Beide Techniken nutzen künstliche Selektion und basieren auf phänotypischen Übergängen. Eine der Techniken beinhaltet außerdem eine Feedback-Schleife zwischen dem metabolischen Phänotyp einer mikrobiellen Population und ihrer chemischen Umgebung.

Curriculum Vitae

Surname: WRIGHT

First name: Jeremiah

Date of birth: 18 May 1982

Nationality: United States of America

Education:

University of Zurich. Zurich, Switzerland. 2006-2011.
Doctorate

University of New Mexico. Albuquerque, New Mexico, USA. 1999-2004.
Bachelor of Science, Chemistry

San Jon High School. San Jon, New Mexico, USA. 1996-1999.
High school diploma

Diploma subject: Systems biology and industrial microbiology

Diploma thesis title: The Acquisition of Valued Phenotypes

Employment:

Network Information Systems Analyst/Programmer II at the Long Term Ecological Research Network Office, University of New Mexico, 2010-Present.

Ph.D. student at the University of Zurich from 2006-2009.

Table of Contents

Article 1

Introduction

Introduction

All organisms possess a set of physical traits, and the *phenotype* is the subset of those traits that are observable [1]. The phenotypic traits of some organisms are valued by people, that is, those traits provide a means to an end, with the ultimate end being satisfaction [2]. The phenotypic traits of an organism might satisfy an esthetic desire – e.g. the color of a flower; a utilitarian desire – e.g. a consumers' good, like food, clothing, or shelter; an intellectual desire – e.g. to test a scientific hypothesis; or any other desire one may have. There are essentially two known biological techniques for acquiring valued phenotypes – *artificial selection*, where the propagation of particular existing phenotypes is encouraged, and *induced phenotypic transition*, where new phenotypes are produced from existing phenotypes. The work described in this thesis is directly related to improving our ability to acquire organisms with valued phenotypes using both of these techniques.

The following perspective is used to discuss those techniques herein. The acquisition of a material *good*, such as a phenotype, is an act of *production*, that is, the transformation of inherently scarce natural resources into more highly valued forms [2]. The application of a particular transformative process, however, depending on both the circumstances and the observer, can be either productive or *destructive*, that is, "goods" (or in this case, *bads*) can be created that have a lower value than the resources consumed during their creation. Therefore, the decision to use a particular technique and the utility it ultimately provides are based on an assessment and comparison of values, that is, *economic analysis* [2, 3]. Thus, throughout this article, I incorporate economic terms and concepts to connect the techniques I discuss to the encompassing concept of production, which is crucial for their beneficial application.

Artificial selection

Selection is any process that favors or induces the survival or reproduction of subpopulations or individuals (subpopulations of size one) over others. Herein, the term *artificial selection* is defined as selection resulting from *human action* (intentional human behavior) [2], with all other instances termed *natural selection*. Thus, selection resulting from *human activity* (unintentional human behavior) is also considered natural. Consequently, artificial selection can only act on phenotypic (observable) traits, because unobservable traits presumably cannot be the focus of human action. Additionally, if artificial selection is attempted, but a phenotype is selected other than the one desired, this is also considered natural selection, because errors are always unintentional [2].

One of the most important aspects of artificial selection is the isolation of particular organisms from a diverse group. It can be reasoned that no two organisms are identical, because at the very least, their coordinates in 3-dimensional space differ (otherwise they would be considered the same organism). Thus, physical diversity among organisms is *always* present. This diversity is also far richer than mere location if one considers the enormous number and types of atoms, molecules, and macromolecules present in all known forms of life. It is difficult to believe that two organisms could be composed of the *exact* same number and types of atoms in the *exact* same configurations and energetic states *ever*, let alone at any given instant. Due to limited measurement capabilities, it is also impossible to accurately observe all of the physical traits of an organism simultaneously. Thus, some subsets of physical traits cannot be elements of the phenotype simultaneously, and the differences in traits among two organisms can never be fully assessed. Therefore,

physical diversity is always present among organisms, and the full extent of that diversity is unknowable.

For macroscopic organisms, artificially selecting subpopulations with desired phenotypes is relatively straightforward, although capturing some wild animals and breeding them in captivity, for example, can require substantial skill and ingenuity. Artificially selecting particular microorganisms, which are difficult to see and handle individually, poses a different set of challenges. One technique for selecting microbes with valued phenotypes is the use of *selective growth conditions*, where the reproduction of certain microbes can be favored (or discouraged) based on their ability to grow in certain environments. If one subpopulation can reproduce more quickly in a particular environment than the others (and their offspring are also considered to be part of that subpopulation), it will become a larger fraction of the population over time. In the extreme (and usually ideal) case, the desired microbes will be the only subpopulation capable of reproduction. The challenge of this technique is to design environments that effectively favor the microbes with the desired properties, which is the focus of the research described in Article 4.

Artificial selection is a powerful tool for acquiring desired phenotypes, but it also has a fundamental limitation – it cannot be used to acquire a phenotype that is not already present. The following section discusses ways that new phenotypes can be produced from existing phenotypes.

Induced phenotypic transitions

It has been theorized that life can emerge from non-living matter, although to the best of our knowledge, this has never been observed by a person in nature or demonstrated

in a laboratory [4, 5]. Rather, in all known cases, new phenotypes are produced through modification of existing phenotypes, a process termed herein as *phenotypic transition*. Specifically, any and all observable changes in the physical characteristics of an organism are regarded here as phenotypic transitions. In this section, some of the known causes of phenotypic transitions are discussed, as well as the ways in which those transitions can be induced by people to produce organisms with valued phenotypes.

Before addressing these topics, a clarification of terminology is necessary to avoid confusion. The words 'gene' and 'genotype' were first published in 1909, when 'gene' was defined abstractly as 'the hereditary unit' and 'genotype' was used primarily as a counterpart to 'phenotype', but without a clear definition provided [1]. In 1954, Watson and Crick proposed that DNA is the macromolecule primarily responsible for the inheritance of traits, and the term 'gene' subsequently became defined more concretely as a sequence of DNA that can be *transcribed* into mRNA and then *translated* into a protein. This new definition of 'gene', however, is not necessarily compatible with its original meaning. Plasmids, for example, are circular pieces of DNA that contain (concrete) genes, yet some plasmids under some conditions are not always transmitted from parent to offspring [6], that is, they do not always function as 'hereditary units'. Conversely, many molecules other than DNA *are* transmitted from parent to offspring, some of which can drastically affect the phenotype and play a crucial role in the inheritance of some traits (see discussion below). Thus, molecules or macromolecules other than DNA can sometimes function as 'hereditary units'. The word 'genotype' has retained its original usage in some contexts, and it has also acquired new meanings. It can be used very specifically to refer to particular allele combinations, or more generally as the 'genetic constitution' of an organism, although this

is quite similar to the current meaning of ‘genome’ – the set of all DNA (not just genes) in an organism. To avoid ambiguity, the word ‘gene’ will always be used here in the modern concrete sense, and ‘genotype’ will not be used at all, except in the remaining discussion, where it is considered to mean *the set of all hereditary units*, which is consistent with the ‘genetic constitution’ definition if the root word ‘gene’ is taken to mean ‘hereditary unit’.

This semantic analysis can be used to make the following point. When ‘genotype’ and ‘phenotype’ are used as counterparts, it is often implicitly assumed that the sets they refer to are disjoint – an element of the genotype cannot be an element of the phenotype, and vice versa. On the contrary, however, if a hereditary unit is observable, then it is an element of *both* sets. Since modern technology enables the sequencing of genes, they are now also elements of the phenotype. Thus, the genotype and phenotype intersect. In the following sections, the known causes of phenotypic transitions are discussed using this perspective.

Genetic causes of phenotypic transitions

The causes of phenotypic transitions can be classified as either genetic or non-genetic. A genetic cause is one in which the DNA sequence of the organism is modified, i.e. a mutation occurs. If a mutation is observable, using a technique such as DNA sequencing, it qualifies as a phenotypic transition, even if all other aspects of the phenotype remain unaltered. Many mutations can, however, have dramatic affects on other parts of the phenotype; thus, if a valued phenotype is not currently present, a mutation might be able to produce it. Since mutations occur naturally, one option is to wait for the right mutation(s) to appear. Techniques to speed this process might be desirable, however, because *time*

preference (i.e. impatience) is an important aspect of human action and production [2]. Mutagenesis and genetic engineering are two such methods.

Mutagenesis is a technique in which organisms are exposed to particular agents (mutagens) that increase the frequency of mutations beyond what would otherwise exist. Radiation, chemicals, and heat have mutagenic affects, and they may act both directly and indirectly to cause mutations. Ultraviolet (UV) radiation is absorbed directly by cellular DNA, which can result in a chemical modification of nucleotides that prevents transcription of the affected gene. UV radiation can also cause mutations indirectly by producing intracellular free radicals, which are known to destroy numerous cellular components, including DNA [7]. Certain alkylating agents, like ethyl methanesulfonate (EMS), have been shown to increase mutation rates in a wide variety of organisms. Mechanisms have been proposed whereby the alkylating agents directly modify the chemical structure of the nucleotides in DNA, or they may indirectly cause mutations by chemically modifying chromosomal proteins, resulting in chromosomal breakage [8]. Exposing some organisms to transient temperature shifts is also known to increase mutation rates, which might be caused by heat-induced deamination of cytosine in DNA to form uracil, or from perturbation of the enzymes involved in DNA synthesis [9].

Genetic engineering is an alternative to mutagenesis, where specific mutations are induced in a more controlled fashion, and many techniques have been developed to accomplish this in various organisms. Genetic transformation is a process in which individual cells acquire foreign DNA from their environment and incorporate it into their genome. For many bacteria, transformation appears to occur commonly in nature, and for many other organisms, it can be induced in a laboratory. By manipulating the genetic content (DNA

sequence) of these vectors, the genome of an organism can be modified with relative precision, although not necessarily permanently. One category of vector, called *plasmids*, can either integrate directly into chromosomes, or remain chromosomally independent (episomal) - replicating autonomously and typically achieving multiple copy numbers within a single cell. Some episomal plasmids have centromeres, and thus segregate with the chromosomes during cell division, but those without centromeres are inherited stochastically, sometimes with a strong maternal bias. Cells can even lose their plasmids entirely (or their offspring may not receive plasmids) and all of the genes they contain. The degree of plasmid instability and its causes depend on the plasmid, the host organism, and the environment [6], and these factors must be considered whenever plasmids are used to induce valued phenotypic transitions.

Not all cells are known to be capable of genetic transformation, so other techniques for introducing genes have been developed, such as *viral transduction*. To reproduce, a virus infects a cell with its small genome, causing the host to express viral genes, which ultimately produces more viruses and sometimes also kills the host. Some researchers try to exploit this naturally occurring gene-delivery and -expression system to induce valued phenotypic transitions, for instance, to cure human genetic disorders using *gene therapy*. To treat diseases using this technique, patients are purposefully infected with viruses whose genomes have been highly modified, sometimes containing < 5% of the original genetic material [10, 11]. This genetic modification typically involves the removal of genes thought to be involved in pathogenesis, the removal of genes thought to trigger the human immune system (for the purpose of evading it), and the addition of potentially therapeutic genes [10].

An alternative to viral transduction is the use of *gene guns* or *biolistics*, where inert nanoparticles, such as gold beads 1 μm in diameter, are coupled with DNA, such as plasmids, and then shot into cells or tissue at high velocity. Some of the particles penetrate the cells and their nuclei, and the attached DNA can then become incorporated into the genome and expressed [12]. Gene guns were used to achieve the first genetic transformations of chloroplasts, mitochondria, and corn (maize), and they have also been used to transform numerous other organisms, including mammals, fungi, bacteria, and algae. Interestingly, the first grant application to develop biolistics technology received "laughter and ridicule" from the evaluating panel, although the grant was ultimately approved [13].

To summarize, genetic mutation can be both an example and a cause of phenotypic transitions, and techniques are available to induce mutations in a large number of organisms, including humans, with varying degrees of control. The choice of technique can be influenced by the degree of understanding of the biochemical processes to be manipulated and, like all other attempts at production, on the perceived costs, benefits, and risks of a particular approach.

Non-genetic causes of phenotypic transitions

There are many examples of non-genetic causes of phenotypic transitions, that is, those that do not involve mutation. One was mentioned above, where a grant application induced a temporary transition of the reviewers' phenotypes to a psychological and behavioral state (i.e. a *mood*) of shared amusement or hostility. Mood can be considered a phenotypic trait, because most people can (at least partially) observe their own mood and the mood of others [14-16]. A correlation appears to exist between certain moods and

altered activity of the limbic and paralimbic systems of the mammalian brain [17, 18], which is also the region that might be responsible for motivation [19] – the prerequisite for goal-directed behaviors, such as human action and production. Mood and its transitions might even influence patterns of production, consumption, and destruction – more phenotypic traits – because it affects one's values [17].

Another non-genetic cause of phenotypic transition is *epigenetics*, which has been defined in many ways [20, 21], but here are three that are currently used, from broad to narrow: 1) the transmission and perpetuation of information through meiosis or mitosis that is not based on the sequence of DNA, 2) meiotically and mitotically heritable changes in gene expression that are not coded in the DNA sequence itself, and 3) the mechanism for the stable maintenance of gene expression that involves physically 'marking' DNA or its associated proteins [22]. One case that satisfies all of these definitions is DNA methylation, wherein nucleotides are chemically modified by the enzymatic addition of a methyl group. These methyl groups can influence the binding of DNA-associated proteins, which can influence, or even regulate, the rate of gene transcription, thereby potentially affecting other aspects of the phenotype. DNA methylation patterns can change over time, but they can also persist inter-generationally, because a methylated strand of DNA can act as a guide for methylation of the (unmethylated) strands that are synthesized during cell division [23].

Epigenesis is also sometimes cited as an example of epigenetics [20], but this is potentially misleading. Epigenesis is the process by which a single (totipotent) cell differentiates into an organism composed of multiple cells and cell types. Since the genomes of these cells are "identical" (highly similar), their phenotypic differences are presumably not attributable to genetic differences, which is indeed somewhat similar to epigenetics. A

central theme of epigenesis, however, is that the offspring *do not* inherit the phenotype of their parent, they acquire a *different* phenotype, which contrasts with epigenetics, where offspring *do* inherit their phenotype, in a way that depends on non-genetic factors. Cellular differentiation can also be strongly influenced by intercellular signal transduction [24]. Thus, the *potential* for differentiated phenotypes is inherited (genetically and possibly also epigentically), but the actual phenotypes can be strongly and dynamically influenced by environmental stimuli, which is further removed from the current idea of epigenetics, at least as it is defined above.

Signal transduction is another non-genetic cause of phenotypic transition in which cells respond to stimuli in their environment through a complex network of cause and effect. One example is when an extracellular chemical binds to a receptor protein located in the cellular membrane, which triggers an intracellular chemical cascade - potentially involving interactions between proteins, ions, small molecules, DNA, and other cellular components - culminating in a change in gene expression. Chemical “pathways” or “circuits” of this nature can sometimes process signals in very complex ways, depending on the strength and pattern (i.e. topology) of their interactions [25]. Particular patterns can, for instance, enable noise filtering, ultrasensitive detection, and the logic operations ‘AND’ and ‘OR’ [26, 27].

A specific example of an interaction pattern is a loop, or structural cycle (not to be confused with a functional cycle, where a value oscillates with time). Article 3 describes an algorithm for the identification of loops in chemical reaction networks. Loops can produce *feedback*, where the output of a process is returned as input, enabling self-monitoring and -regulation. In simple cases, feedback can either increase or decrease signal sensitivity,

response time, and response variability [24, 25]. In more complex cases, feedback systems can even mimic the behavior a toggle switch, where a system can be signaled to transition between two stable (digital) states and then remain in that state after the stimulus is removed [28]. The latter is a characteristic of memory, and such phenomena might explain why cellular differentiation is "irreversible," that is, differentiated cells do not appear to un-differentiate [24, 28]. Feedback can be intra- and intercellular, and between organisms and their environment - an example of which is discussed in Article 4, where feedback was utilized to artificially select cells with valued phenotypes.

Some of these non-genetic phenomena might be exploited to induce desirable phenotypic transitions. Genetic engineering has been used, for example, to introduce chemical interaction patterns into microbial cells to demonstrate their signal processing capabilities [29-32]. There are also many cases where the environment can be modified to induce desirable phenotypic transitions. The yeast *Saccharomyces cerevisiae*, for example, produces ethanol in some environments, but not others [33]. Therefore, if ethanol production is a valued phenotype, but it is not currently present in a population of *S. cerevisiae*, the environment can be modified to induce that phenotypic transition, such as by adding fermentable carbon sources and by removing fermentation inhibitors, like oxygen [34, 35]. Non-genetic factors might, however, also interfere with, or even prevent, the desired effects of genetic engineering, especially if their existence is unknown or ignored.

To summarize, successfully inducing desired phenotypic transitions, either genetically or non-genetically, can require substantial knowledge of intracellular processes and the ways in which organisms interact with their environment. These are the primary topics of interest in the field of study known as *systems biology*.

Systems biology

Organisms are typically composed of a vast number of interacting elements of different types – e.g. DNA, RNA, proteins, metabolites, lipids, carbohydrates, ions, metals, organelles, cells, tissues, and organs. Systems biology is devoted to exhaustively studying these components, their interactions with each other, and their interactions with their environment. If these factors fully determine the physical traits of organisms, then systems biology can be appropriately called the study of phenotypes. The interested reader can find abundant literature regarding systems biology.

Since the number of components and interactions in a single organism is extremely large, computers and software are central and essential tools in systems biology. Even very simple forms of system-level data acquisition, storage, and analysis would be nearly impossible without them. To address the computational demands of systems biology research, I developed a software package called the *Systems Biology Research Tool*, which provides standardized formats for storing data, implementations of numerous algorithms for analyzing and simulating biological systems, and mechanisms for adding new computational techniques as the need arises. The features and capabilities of the Systems Biology Research Tool are described more thoroughly in Article 2. The most prevalent technique implemented in the Systems Biology Research Tool is *flux balance analysis*, which is used to simulate the behavior of networks of chemical reactions.

Flux balance analysis

Cellular systems and processes can be described in terms of the chemical reactions that occur within them. During a chemical reaction, one set of chemical species, the

reactants or *substrates*, is transformed into another set of chemical species, the *products*. The conversion of reactants to products appears to always occur in fixed proportions, that is, constant ratios can be determined for each species involved. In the reaction 3 A + 2 B → 1 D, for example, the coefficients 3, 2, and 1 indicate the conversion ratios, called *stoichiometric coefficients*. Chemical reactions must also be *balanced*, that is, the number and types of atoms in the reactants must be identical to those in the products. These properties constrain the ways in which chemical reactions function.

A set of chemical reactions can be represented using a *stoichiometry matrix*, where each column corresponds to a single reaction, each row to a single chemical species, and each element of the matrix indicates the stoichiometric coefficient of the respective species in the respective reaction. Thus, a stoichiometry matrix is used to record some of the constraints on a chemical reaction system, such as mass conservation (assuming the reactions are properly balanced). The direction of a reaction (as it is written) can also be preserved in a stoichiometry matrix by recording the stoichiometric coefficients of reactants and products as negative and positive numbers, respectively. Stoichiometry matrices are central to many methods of chemical network analysis [36].

In flux balance analysis, chemical reactions are classified as either *internal* or *exchange* reactions. Internal reactions are those comprising the system being studied, and exchange reactions are pseudo-reactions that are used to supply (remove) chemical species to (from) the reaction system, that is, they allow the system to interact with its surroundings. The flux, or rate, of each internal and exchange reaction can also be constrained to lie within an interval, such as [0, ∞), which, in this case, indicates that the reaction can occur at any rate, but only in the “forward” direction. These reactions are used

to construct a stoichiometry matrix *S* which is used to formulate the equation $Sv = 0$, where *v* is a vector of fluxes (velocities) of all reactions in the network. This equation imposes another constraint on the reaction system, called the "steady-state constraint," because it requires that the concentrations of chemical species in the system remain constant. The solutions to this equation that satisfy all of the defined constraints are the allowable states of the network [37].

The set of all allowable network states is called the *flux space*, which is a geometric object. Flux spaces often have a particular size and shape, although empty flux spaces can also exist, if their constraints are defined such that no flux vector can possibly satisfy them. Two basic methods can be used to explore a flux space - biased and unbiased sampling. By applying linear programming, for example, the rates of particular reactions, or linear combinations of reaction rates, can be optimized (minimized or maximized) to achieve a biased sampling. In the standard version of flux balance analysis, where all of the constraints are linear, the flux space is a convex polytope, and the optimal flux vectors lie only on the edges [37]. Another method is to sample randomly from the interior of flux space. A large set of randomly sampled flux vectors can theoretically be used to determine the size (hypervolume) and shape of the flux space, which is one measure of a network's capabilities [38-40]. Both biased and unbiased sampling techniques are implemented in the Systems Biology Research Tool.

Flux balance analysis, however, can produce results (flux vectors) that are inconsistent with the laws of thermodynamics, because linear constraints are sometimes insufficient for that purpose [41, 42]. If the appropriate nonlinear constraints are used, however, the time required to compute a solution increases exponentially with the problem

size, making some networks impossible to analyze [43]. One approach to this problem is to generate a set of unbiased flux vectors within a standard flux space, and then eliminate the vectors that are unrealistic [41]. This method relies on the identification of particular cycles in the reaction network, although, the computation time of cycle identification can also scale exponentially with problem size [44, 45]. In Article 3, I describe an algorithm that can dramatically reduce the computation time and memory usage of cycle identification for some genome-scale metabolic networks, which are currently analyzed quite frequently, especially for the purpose of acquiring valued microbial phenotypes [46].

Genome-scale metabolic networks

Metabolism is the process by which organisms acquire matter and energy from their environment - using a large set of chemical reactions - to live, grow, and reproduce. Since many organisms, especially microbes, are valued for their ability to consume and produce certain chemicals, a better understanding of metabolism might be directly applicable to the acquisition of valued phenotypes [47]. An effort in this direction is the reconstruction of genome-scale metabolic networks, where an attempt is made to document all of the chemical species, reactions, enzymes, and genes that are responsible for an organism's metabolic functions. The first eukaryotic organism to have its metabolic network reconstructed was the unicellular yeast *Saccharomyces cerevisiae*, which is a model organism in biological research and is used to make bread, beer, and wine. That reconstruction process, which was a collaborative effort between two research groups [48], is summarized below.

The metabolic model (i.e. the reaction list and associated metadata) of *S. cerevisiae* was compiled from various online databases, textbooks, and journal articles. The absence of

necessary information - such as cofactor requirements of enzymes, intracellular localization of reactions, or reaction direction - was managed by applying explicit rules or by using inference. A pseudo-reaction was also added to the model to enable the simulation of cell growth using flux balance analysis. To simulate aerobic growth on a minimal glucose medium, for example, oxygen, glucose, ammonia, phosphate, and hydrogen sulfide were provided in the simulated environment, and the reaction network was then responsible for transporting these nutrients into the cell and transforming them into the molecular components necessary for biomass production. Initially, however, no agreement between computed and experimental results could be found [48]. Consequently, reactions were added, inactivated, made irreversible, or otherwise adjusted until the simulated results agreed with the empirical data. Thus, this iterative reconstruction process is another example of feedback, where the performance of the network model motivated further modification of the model to alter its performance. The final network, called iFF708, contained 1175 metabolic reactions, 584 metabolites, and 708 open reading frames (sections of the genome that potentially encode for proteins), to which 1035 reactions were assigned via their catalyzing enzyme [48]. Other metabolic network models of this nature have also been constructed for *S. cerevisiae* and numerous other organisms [46].

Models with this level of granularity enable the simulation of the addition and removal of individual components from organisms (genes, proteins, and chemical reactions) and their environment (chemical species). Several points can be made from and about this type of analysis. 1) An organism's metabolic capabilities are determined interdependently by its components and environment, i.e. a metabolic pathway cannot function if the environment lacks a necessary substrate or if the organism lacks a necessary component. 2)

An organism's metabolic activity alters its environment, by consuming and producing chemicals, and an organism's environment influences its metabolic activity, which together allows feedback. 3) If these models provide adequate predictive ability, they can be used to acquire organisms with valued metabolic phenotypes, since phenotypic transitions can be induced by modifying either the environment or the organism, and artificial selection can be achieved by creating environments that favor particular phenotypes (see above).

In the preceding sections, the use of artificial selection and induced phenotypic transitions were described as independent methods for acquiring valued phenotypes. The following section describes combinations of these techniques, primarily as they relate to the acquisition of organisms with valued metabolic phenotypes.

Combining artificial selection with induced phenotypic transitions

A classic example of combining artificial selection with induced phenotypic transitions is the practice of selective breeding, wherein two organisms are bred with the intent of producing offspring with valued phenotypes. In some cases, the intent is only to preserve the valued traits of one or both of the parents, but in others, the intent is to combine distinct traits of both parents to produce offspring with a new phenotype. The latter is an example of an intergenerational phenotypic transition.

Selective breeding is also an example of economic production, where goods are derived (ultimately) from natural resources. When analyzing production, a distinction can be made between goods used for immediate consumption ("consumers' goods") and those used during the production of other goods ("producers' goods") [2]. In selective breeding, the parents are producers' goods, and the offspring are either producers' or consumers'

goods, depending on their ultimate use. An example is described below where selective breeding was used to produce a strain of *S. cerevisiae* for use (as a producers' good) in the production of bioethanol from lignocellulose, i.e. plant material.

Lignocellulose is composed primarily of the polymers cellulose, hemicellulose, and lignin; whose proportions vary depending on their origin, e.g. grasses, hard or soft woods, etc. Cellulose is a carbohydrate composed of glucose, which can be readily fermented by numerous species of microorganisms. Hemicellulose and lignin are both irregular polymers whose compositions also vary depending on their origin. Lignin primarily contains various aromatic residues, for which relatively few microbes are known to be capable of their fermentation or degradation. Hemicellulose is a carbohydrate composed primarily of pentose monomers, such as xylose and arabinose, which can also be fermented by various bacteria and fungi, although the commonly used industrial yeast *S. cerevisiae* is known to be poor in its ability to metabolize the pentose sugars that are predominant in hemicellulose [34, 49].

S. cerevisiae is already widely used for industrial ethanol production (from non-lignocellulosic feedstocks), but native strains are only known to grow extremely slowly using xylose as a carbon source [50]. To produce ethanol from lignocellulose using *S. cerevisiae*, a strain is required that is capable of fast xylose utilization (due, at least partially, to time preference [2]). To acquire such a strain, the company Microbiogen attempted a selective breeding program, wherein various industrial, laboratory, and wild-type strains of *S. cerevisiae* were co-cultured under selective growth conditions for 4 years [50]. That culture was subjected to 23 mating cycles over that period, where cells were sporulated, germinated, and mass-mated to produce a genetically and phenotypically heterogeneous

population. At the end of that process, 30 individual strains were isolated, and their ability to utilize xylose for growth was characterized. The doubling time of these strains ranged from 5 to 8 hours, compared to the initial strains for which growth was nearly undetectable [50]. By combining artificial selection and induced phenotypic transitions in this way, the desired phenotype was acquired.

Another example of combining artificial selection and induced phenotypic transition is the practice of genetically engineering microbes and then subjecting them to long-term cultivation under selective growth conditions. This technique was applied to a laboratory strain of *S. cerevisiae* (CEN.PK113-7D), also for the purpose of acquiring a phenotype capable of fast xylose utilization. The expression of five existing genes was increased, one gene was deleted, and a new gene (from the fungus *Piromyces sp.*) was added to the genome to produce a strain called RWB217. This strain had a specific growth rate of 0.09 h^{-1} in xylose supplemented synthetic medium, whereas growth was undetected for CEN.PK113-7D [51]. RWB217 was then subjected to 2,100 hours (87.5 days) of cultivation in two separate conditions designed to select cells with the best ability to utilize xylose. During those cultivations, phenotypic transitions were observed at the population level, which resulted either from naturally occurring genetic or non-genetic causes, or a combination of both. Isolates were obtained at the end of cultivation, and one of these was termed RWB218, which exhibited an even higher specific growth rate of 0.12 h^{-1} [52]. Again, the desired phenotype was acquired by combining artificial selection with both induced and non-induced phenotypic transitions.

Fast xylose utilization alone, however, is not sufficient for the industrial production of bioethanol from lignocellulose. When lignocellulose is hydrolyzed (a prerequisite for

fermentation), the resulting hydrolysate can contain a variety of fermentation inhibitors [34, 49, 53], such as acetic acid (which is used as a food preservative for that very reason [54]). In fact, the concentrations of xylose and acetic acid in hydrolysates are potentially coupled, because xylose is present in hemicellulose primarily as a polymer called xylan, which can be highly acylated [55]. Upon hydrolysis, the xylose units and acyl groups of xylan are cleaved, resulting in free xylose and acetate molecules [49, 55]. Therefore, fast xylose utilization in the presence of acetic acid is another apparent requirement for the fermentation of lignocellulosic hydrolysates. Article 4 discusses additional research to acquire such a phenotype.

Concluding remarks

The focus of this article was on the acquisition of valued phenotypes using artificial selection and induced phenotypic transitions, which are both examples of economic production. A large fraction of modern economic production is devoted to supporting human metabolic activity, e.g. providing food, clothing, shelter, energy, etc.; and those processes - metabolism and economic production - are very similar, both involving the transformation of naturally occurring resources into “products”, to sustain life and provide satisfaction, respectively [56]. Psychological sensations, such as satisfaction, are also associated with biochemical events, e.g. signals from the environment are transmitted and processed by a network of chemical and cellular interactions (the nervous system) to ultimately produce perceptions. These phenomena are all examples of phenotypic transitions, to the extent that they are observable. The work described in this dissertation is a small contribution to understanding these subjects and their interrelations.

Scope of this dissertation

The preceding sections provide the background information required to understand the context of the work presented in this dissertation. Article 2 describes a free, open-source software platform called the Systems Biology Research Tool, which I developed to improve our ability to understand the interdependence of organisms, their components, and their environment. Article 3 describes an algorithm for exhaustively identifying cycles in large chemical reaction networks. Knowledge of these cycles can be used to correct thermodynamically unrealistic predictions from flux balance analysis, and to identify potential feedback loops in cellular processes. Article 4 describes the results of two techniques involving long-term cultivation of microbes under selective growth conditions to acquire a valued phenotype – fast xylose consumption by *S. cerevisiae* in the presence of acetic acid, a fermentation inhibitor. Since techniques from both computational systems biology and experimental microbiology are described in the following articles, a deliberate attempt was made to make this work accessible and understandable to people working in either (or neither) of those fields. This is important for an effective exchange of ideas and for becoming familiar with different schools of thought, which are both essential in multidisciplinary research.

References

1. J.H. Wanscher. The history of Wilhelm Johannsen's genetical terms and concepts from the period 1903 to 1926. *Centaurus*, 19(2):125–147, 1975.
2. M.N. Rothbard. *Man, Economy, and State with Power and Market*. The Ludwig von Mises Institute, 2009.
3. J.A. Schumpeter. *The theory of economic development: an inquiry into profits, capital, credit, interest, and the business cycle*. Transaction Publishers, 2008.
4. J. Oro, S.L. Miller, and A. Lazcano. The origin and early evolution of life on Earth. *Annual Review of Earth and Planetary Sciences*, 18(1):317–356, 1990.
5. L.E. Orgel. The origin of life–a review of facts and speculations. *Trends in Biochemical Sciences*, 23(12):491–495, 1998.
6. M.A. Romanos, C.A. Scorer, and J.J. Clare. Foreign gene expression in yeast: a review. *Yeast*, 8(6):423–488, 1992.
7. R.P. Sinha and D.P. Häder. UV-induced DNA damage and repair: a review. *Photochemical & Photobiological Sciences*, 1(4):225–236, 2002.
8. G.A. Sega. A review of the genetic effects of ethyl methanesulfonate. *Mutation Research/Reviews in Genetic Toxicology*, 134(2-3):113–142, 1984.
9. J.W. Drake and R.H. Baltz. The biochemistry of mutagenesis. *Annual Review of Biochemistry*, 45(1):11–37, 1976.
10. C.E. Thomas, A. Ehrhardt, and M.A. Kay. Progress and problems with the use of viral vectors for gene therapy. *Nature Reviews Genetics*, 4(5):346–358, 2003.
11. E. Poeschla, P. Corbeau, and F. Wong-Staal. Development of HIV vectors for anti-HIV gene therapy. *Proceedings of the National Academy of Sciences of the United States of America*, 93(21):11395, 1996.
12. J.C. Sanford. The biolistic process. *Trends in Biotechnology*, 6(12):299–302, 1988.
13. J.C. Sanford. The development of the biolistic process. *In Vitro Cellular & Developmental Biology-Plant*, 36(5):303–308, 2000.
14. R.E.A. Green, G.R. Turner, and W.F. Thompson. Deficits in facial emotion perception in adults with recent traumatic brain injury. *Neuropsychologia*, 42(2):133–141, 2004.
15. S. McDonald and S. Flanagan. Social perception deficits after traumatic brain injury: interaction between emotion recognition, mentalizing ability, and social communication. *Neuropsychology*, 18(3):572, 2004.
16. K. Kucharska-Pietura, M.L. Phillips, W. Gernand, and A.S. David. Perception of emotions from faces and voices following unilateral brain damage. *Neuropsychologia*, 41(8):1082–1090, 2003.
17. R.R. Prechter. *The wave principle of human social behavior and the new science of socionomics*. New Classics Library, 2002.
18. D.D. Dougherty, L.M. Shin, N.M. Alpert, R.K. Pitman, S.P. Orr, M. Lasko, M.L. Macklin, A.J. Fischman, and S.L. Rauch. Anger in healthy men: a PET study using script-driven imagery. *Biological Psychiatry*, 46(4):466–472, 1999.
19. G.J. Mogenson, D.L. Jones, and C.Y. Yim. From motivation to action: functional interface between the limbic system and the motor system. *Progress in Neurobiology*, 14(2-3):69–97, 1980.
20. A. Bird. Perceptions of epigenetics. *Nature*, 447(7143):396–398, 2007.
21. A.D. Goldberg, C.D. Allis, and E. Bernstein. Epigenetics: a landscape takes shape. *Cell*, 128(4):635–638, 2007.
22. J.M. Levenson and J.D. Sweatt. Epigenetic mechanisms in memory formation. *Nature Reviews Neuroscience*, 6(2):108–118, 2005.

23. J. Bender. DNA methylation and epigenetics. *Annual Review of Plant Biology*, 55:41, 2004.

24. M. Freeman. Feedback control of intercellular signalling in development. *Nature*, 408(6810):313–319, 2000.

25. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

26. S. Hooshangi, S. Thiberge, and R. Weiss. Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. *Proceedings of the National Academy of Sciences of the United States of America*, 102(10):3581, 2005.

27. U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

28. J.E. Ferrell. Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability. *Current Opinion in Cell Biology*, 14(2):140–148, 2002.

29. S. Mangan, S. Itzkovitz, A. Zaslaver, and U. Alon. The incoherent feed-forward loop accelerates the response-time of the gal system of *Escherichia coli*. *Journal of Molecular Biology*, 356(5):1073–1081, 2006.

30. S. Mangan, A. Zaslaver, and U. Alon. The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. *Journal of Molecular Biology*, 334(2):197–204, 2003.

31. A. Becskei and L. Serrano. Engineering stability in gene networks by autoregulation. *Nature*, 405(6786):590–593, 2000.

32. T.S. Gardner, C.R. Cantor, and J.J. Collins. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403(6767):339–342, 2000.

33. E. Postma, C. Verduyn, W.A. Scheffers, and J.P. Van Dijken. Enzymic analysis of the Crabtree effect in glucose-limited chemostat cultures of *Saccharomyces cerevisiae*. *Applied and Environmental Microbiology*, 55(2):468, 1989.

34. A.J.A. van Maris, D.A. Abbott, E. Bellissimi, J. van den Brink, M. Kuyper, M.A.H. Luttik, H.W. Wisselink, W.A. Scheffers, J.P. van Dijken, and J.T. Pronk. Alcoholic fermentation of carbon sources in biomass hydrolysates by *Saccharomyces cerevisiae*: current status. *Antonie van Leeuwenhoek*, 90(4):391–418, 2006.

35. J.P. Dijken, R.A. Weusthuis, and J.T. Pronk. Kinetics of growth and sugar consumption in yeasts. *Antonie van Leeuwenhoek*, 63(3):343–352, 1993.

36. A.K. Gombert and J. Nielsen. Mathematical modelling of metabolism. *Current Opinion in Biotechnology*, 11(2):180–186, 2000.

37. C.H. Schilling, D. Letscher, and B.Ø. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, 203(3):229–248, 2000.

38. S.J. Wiback, I. Famili, H.J. Greenberg, and B. Ø. Palsson. Monte Carlo sampling can be used to determine the size and shape of the steady-state flux space. *Journal of Theoretical Biology*, 228(4):437–447, 2004.

39. N.D. Price, J. Schellenberger, and B.Ø. Palsson. Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies. *Biophysical Journal*, 87(4):2172–2186, 2004.

40. J. Schellenberger and B.Ø. Palsson. Use of randomized sampling for analysis of metabolic networks. *Journal of Biological Chemistry*, 284(9):5457, 2009.

41. N.D. Price, I. Thiele, and B.Ø. Palsson. Candidate states of *Helicobacter pylori*'s genome-scale metabolic network upon application of "loop law" thermodynamic constraints. *Biophysical Journal*, 90(11):3919–3928, 2006.

42. D.A. Beard, S. Liang, and H. Qian. Energy balance for analysis of complex metabolic networks. *Biophysical Journal*, 83(1):79–86, 2002.

43. F. Yang, H. Qian, and D.A. Beard. Ab initio prediction of thermodynamically feasible reaction directions from biochemical network stoichiometry. *Metabolic Engineering*, 7(4):251–259, 2005.

44. J.A. Papin, J. Stelling, N.D. Price, S. Klamt, S. Schuster, and B.Ø. Palsson. Comparison of network-based pathway analysis methods. *Trends in Biotechnology*, 22(8):400–405, 2004.

45. J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5(1):175, 2004.

46. M.A. Oberhardt, B.Ø. Palsson, and J.A. Papin. Applications of genome-scale metabolic reconstructions. *Molecular Systems Biology*, 5(1), 2009.

47. G. Stephanopoulos. Metabolic engineering. *Current Opinion in Biotechnology*, 5(2):196–200, 1994.

48. J. Förster, I. Famili, P. Fu, B.Ø. Palsson, and J. Nielsen. Genome-scale reconstruction of the *Saccharomyces cerevisiae* metabolic network. *Genome Research*, 13(2):244, 2003.

49. L. Olsson and B. Hahn-Hägerdal. Fermentation of lignocellulosic hydrolysates for ethanol production. *Enzyme and Microbial Technology*, 18(5):312–331, 1996.

50. P.V. Attfield and P.J.L. Bell. Use of population genetics to derive nonrecombinant *Saccharomyces cerevisiae* strains that grow using xylose as a sole carbon source. *FEMS Yeast Research*, 6(6):862–868, 2006.

51. M. Kuyper, M.M.P. Hartog, M.J. Toirkens, M.J.H. Almering, A.A. Winkler, J.P. Dijken, and J.T. Pronk. Metabolic engineering of a xylose-isomerase-expressing *Saccharomyces cerevisiae* strain for rapid anaerobic xylose fermentation. *FEMS Yeast Research*, 5(4-5):399–409, 2005.

52. M. Kuyper, M.J. Toirkens, J.A. Diderich, A.A. Winkler, J.P. Dijken, and J.T. Pronk. Evolutionary engineering of mixed-sugar utilization by a xylose-fermenting *Saccharomyces cerevisiae* strain. *FEMS Yeast Research*, 5(10):925–934, 2005.

53. C.R. Fischer, D. Klein-Marcuschamer, and G. Stephanopoulos. Selection and optimization of microbial hosts for biofuels production. *Metabolic Engineering*, 10(6):295–304, 2008.

54. P. Piper, C.O. Calderon, K. Hatzixanthis, and M. Mollapour. Weak acid adaptation: the stress response that confers yeasts with resistance to organic acid food preservatives. *Microbiology*, 147(10):2635, 2001.

55. A. Sunna and G. Antranikian. Xylanolytic enzymes from fungi and bacteria. *Critical Reviews in Biotechnology*, 17(1):39–67, 1997.

56. H.E. Daly. On economics as a life science. *The Journal of Political Economy*, 76(3):392–406, 1968.

Article 2

The Systems Biology Research Tool:

evolvable open-source software

Originally published as: J. Wright and A. Wagner. The Systems Biology Research Tool: evolvable open-source software. *BMC Systems Biology*, 2(1):55, 2008.

Abstract

Background: Research in the field of systems biology requires software for a variety of purposes. Software must be used to store, retrieve, analyze, and sometimes even to collect the data obtained from system-level (often high-throughput) experiments. Software must also be used to implement mathematical models and algorithms required for simulation and theoretical predictions on the system-level.

Results: We introduce a free, easy-to-use, open-source, integrated software platform called the *Systems Biology Research Tool* (SBRT) to facilitate the computational aspects of systems biology. The SBRT currently performs 35 methods for analyzing stoichiometric networks and 16 methods from fields such as graph theory, geometry, algebra, and combinatorics. New computational techniques can be added to the SBRT via *process plug-ins*, providing a high degree of evolvability and a unifying framework for software development in systems biology.

Conclusions: The Systems Biology Research Tool is a technological advance for systems biology. This software can be used to make sophisticated computational techniques accessible to everyone (including those with no programming ability), to facilitate cooperation among researchers, and to expedite progress in the field of systems biology.

Background

Some of the primary goals of systems biology are to identify and quantify the individual components of cells, organs, and organisms; to understand the interactions between these components; and to use this information to create mathematical models that enable accurate predictions. Since organisms are composed of large numbers of unique elements (i.e. genes, proteins, metabolites, etc.), and since many interactions often exist between these elements, even the most basic forms of system-level data analysis or simulation cannot be done by hand. Instead, software must be used to store, retrieve, analyze, and sometimes even to collect the data obtained from system-level experiments. Software must also be used to implement mathematical models and algorithms required for simulation and theoretical predictions on the system-level.

We introduce an integrated software platform called the *Systems Biology Research Tool* (SBRT) to facilitate the computational aspects of systems biology. The SBRT is useful for both the management and analysis of data, and the simulation and prediction of cellular phenotypes. The SBRT can, for example, be used to translate data files into various machine- and human-readable formats; to simulate the activity of reconstructed signal transduction and genome-scale metabolic networks using *flux balance analysis* and related methods [1, 2]; and to analyze the topology of experimentally determined biochemical reaction networks, such as transcriptional regulation and protein-protein interaction networks. Since new data formats, methods of data analysis, and simulation techniques arise frequently during systems biology research, the SBRT is also designed to allow independent software developers to add new functionality as it is needed.

Materials and methods

Performance comparisons

Of all existing packages, the COBRA Toolbox is most similar to the SBRT in terms of the computational procedures offered by both (see Results and discussion). Because of these similarities, we performed a comparative performance analysis of some capabilities offered by both packages. Specifically, we carried out 5 analyses using the *in silico* model of *Saccharomyces cerevisiae* metabolism iND750 [17]. In analyses A and B, the model was provided a minimal growth-supporting medium with glucose as the sole carbon and energy source. This was achieved by setting the maximum glucose supply rate to an arbitrary value of 1, constraining the supply rates of oxygen, ammonium, sulfate, phosphate, and water to be unbounded in the positive direction, and setting the supply rates of all other extracellular metabolites to zero. In analyses A and B, the variability of all fluxes in the network and the effect of all single-gene deletions on the maximum biomass production rate were determined, respectively.

In analyses C, D, and E, the iND750 model was sequentially provided 100 randomly generated media. Each of these media was created beforehand by setting the maximum supply rate of 58 (one half of the total) randomly chosen extracellular metabolites to the value 10 and setting the remaining supply rates to zero and by ensuring each medium supported biomass production. Identical sets of media were used in analyses C, D, and E and by both the Systems Biology Research Tool and the COBRAToolbox. In analyses C, D, and E, the maximum biomass production rate, the variability of all fluxes, and the effect of all

single-gene deletions on the maximum biomass production rate were computed, respectively.

Software versions

The Systems Biology Research Tool 1.1.0 and the COBRAToolbox 1.3.3 were used for all performance comparisons. The SBRT ran within Sun's Java HotSpot(TM) 64-Bit Server VM (build 1.6.0_03-b05, mixed mode) and used GLPK 4.8 to solve all linear programs and Xerces-J 2.1.0 to parse SBML files. The COBRAToolbox ran within MATLAB 7.2.0 and used GLPK 4.23 to solve all linear programs and libsbml 2.3.4 to parse SBML files. GLPKMEX 2.4 and the SBMLToolbox 2.0.2 were used by the COBRAToolbox to enable communication with GLPK and libsbml, respectively.

Program execution

All performance comparisons were made on a Dell Precision 490 computer equipped with a 2.33 GHz Intel Xeon processor with Kubuntu 7.10 (AMD64) as the operating system. A bash script was used to execute 10 programs sequentially for each analysis for each software package. The time was recorded before each program began and after each program finished execution to determine the total running time. A perl script (`memmon`) was used to frequently sample the contents of `/proc/meminfo` to monitor the memory usage during each program execution. Memory monitoring began before each program was executed to establish a baseline, and the maximum memory consumption during program execution was measured relative to this baseline.

Source code and data

The scripts, MATLAB m files, and input files used to perform these analyses, and the data generated, are all accessible at the URL `http://www.bioc.uzh.ch/wagner/software/SBRT/suppl_material.zip`.

Results and discussion

Implementation

The SBRT is both an application and an application programming interface (API). It is written in Java and has been tested in Windows XP, Mac OS X, and two distributions of Linux, requiring no modification of source code or recompilation. The SBRT is licensed under the GNU General Public License and is therefore open-source, modifiable, and freely distributable. The most recent versions of the SBRT can be downloaded from the SBRT's homepage [3].

The Systems Biology Research Tool's API contains over 300 well tested and fully documented classes and interfaces. The API is composed of two functionally distinct levels: the *kernel*, which is responsible for performing all significant computation, and the *shell*, which is responsible for relaying information between the user and the kernel. The kernel is completely independent of the shell, which results in a great degree of flexibility and robustness: new functionality can be added to the kernel without concern for user-level I/O details; new functionality can be added to the shell without modifying the kernel, thereby preventing the introduction of kernel-level errors. The kernel contains implementations of algorithms, methodological procedures, and fundamental *objects*, such as networks, chemical reactions, mathematical expressions, matrices, convex polytopes, hyperplanes,

linear program solvers, etc. The shell is primarily composed of classes and interfaces for reading(writing) files from(to) the hard drive, for parsing and formatting various types of data, and for managing and monitoring kernel-level activities.

Use as an application

The SBRT can be used as an application to execute *processes*. A process is a series of actions that takes user-supplied input and produces a result. The SBRT includes 35 processes for analyzing stoichiometric networks, such as optimizing objective functions, computing the variability of fluxes, identifying reaction pathways, generating uniformly distributed points within flux spaces, analyzing the properties of flux vectors and intervals, and more. The SBRT also includes 16 processes utilizing graph theory, geometry, algebra, statistics, and combinatorics. Table 1 contains short descriptions of these 51 processes.

Processes can be controlled with simple text-based input files (that can be created using common word processing or spreadsheet applications) or directly from the command line. When possible, files generated by one process can also be used as *input* files in other SBRT processes, allowing the user to design complex analyses by linking processes via their input and output files, without writing a single line of code. For example, the process *BiGG-SBML File Reader* can be used to translate a machine-readable file into a human-readable and -editable text file *R* that contains a list of chemical reactions. The file *R* can then be supplied to the *Network Information Gatherer* process to create a text file *N* that contains

Table 1. Descriptions of the 51 processes currently implemented in the Systems Biology Research Tool.

Category	Process Name	Brief Description
Flux Optimization	FBA Optimization	Used to compute the optimal value of a flux or linear combination of fluxes in a stoichiometric network.
	Reaction Deletion	Used to compute the effect of deleting sets of reactions in a stoichiometric network.
	Catalyst Deletion	Used to compute the effect of deleting sets of catalysts in a stoichiometric network.
	Objective Function Analysis	Used to compute the optimal values of multiple objective functions for a stoichiometric network.
	Constraint Variation	Used to compute the optimal values of a single objective function for multiple sets of flux constraints.
	Constraint Variation-Reaction Deletion	Used to compute the combined effects of deleting reactions and varying the flux constraints in a stoichiometric network.
	Constraint Variation-Catalyst Deletion	Used to compute the combined effects of deleting catalysts and varying the flux constraints in a stoichiometric network.
	Constraint Variation-Objective Function Analysis	Used to compute the optimal values of multiple objective functions for multiple sets of flux constraints.
Flux Variability	Simple Flux Intervals	Used to compute the intervals of fluxes in a stoichiometric network in the simplest possible way.
	Constrained Reverse Reaction Flux Intervals	Used to compute the intervals of fluxes in a stoichiometric network after constraining the fluxes of reversible reactions.
	Flux Cap Identification	Used to create *caps* for each unbounded flux in a stoichiometric network.
	Mahadevan-Schilling Flux Intervals	Used to compute the Mahadevan-Schilling flux intervals in a stoichiometric network.
	Constraint Variation-Simple Flux Intervals	Used to compute the *simple* flux intervals for multiple sets of flux constraints.
	Constraint Variation-Constrained Reverse Reaction Flux Intervals	Used to compute *constrained reverse reaction* flux intervals for multiple sets of flux constraints.
	Constraint Variation-Mahadevan-Schilling Flux Intervals	Used to compute Mahadevan-Schilling flux intervals for multiple sets of flux constraints.
Chemical Reaction Pathway Identification	Extreme Current Identification	Used to identify the extreme currents in stoichiometric networks.
	WW Network Reduction	Used to reduce the size of stoichiometric networks for the purpose of identifying the cycles they contain.
	MS Network Reduction	Used to reduce the size of stoichiometric networks for the purpose of identifying the cycles they contain.
	SLP Cycle Identification	Used to identify the cycles in stoichiometric networks.
Flux Space Sampling	Random Constraint Generator	Used to generate random flux constraints.
	Random Objective Function Generator	Used to generate random objective functions.
	Initial Point Generator	Used to compute an initial flux vector for use in CD Hit-and-Run Analysis.
	Coordinate Direction Hit-and-Run Analysis	Used to compute random, uniformly-distributed flux vectors from the interior flux space.
	Space Variation-Initial Point Generator	Used to compute initial flux vectors for use in Space Variation-CD Hit-and-Run Analysis.
	Space Variation-Coordinate Direction Hit-and-Run Analysis	Used to compute random, uniformly-distributed flux vectors from the interiors of multiple flux spaces.
Flux Data Analysis	Flux Activity Analysis	Used to analyze the activity of fluxes in a collection of flux vectors.
	Flux Plasticity Analysis	Used to analyze the plasticity of fluxes in a collection of flux interval vectors.

Table 1 (continued). Descriptions of the 51 processes currently implemented in the Systems Biology Research Tool.

Category	Process Name	Brief Description
Stoichiometric Network Utilities	Simple Reaction File Reader	Used to translate files containing a list of chemical reactions into human-readable *FBA Reaction Files*.
	Palsson-SBML File Reader	Used to read SBML files from Dr. Palsson's website.
	BiGG-SBML File Reader	Used to read SBML files from the BiGG Database.
	Palsson-SBML File Translation	Used to translate SBML files from Dr. Palsson's website into human-readable *FBA Reaction Files* and *Reaction-Catalyst Association Files*.
	BiGG-SBML File Translation	Used to translate SBML files from the BiGG Database into human-readable *FBA Reaction Files* and *Reaction-Catalyst Association Files*.
	Metatool File Writer	Used to convert *FBA Reaction Files* into input files for Metatool.
	Network Information Gatherer	Used to gather basic information about a stoichiometric network.
	FBA System Solver	Used to solve the equation $Sv = 0$.
Graph Theory	Path Identification in a Directed Graph	Used to identify the simple paths in a directed graph.
	Cycle Identification in a Directed Graph	Used to identify the simple cycles in a directed graph.
Geometry	Coordinate Directions Hit-and-Run	Used to generate random interior points within convex polytopes.
Algebra	Linear System Solver	Used to solve systems of linear equations using Mathematica.
	Multiple-Vectors File Conversion	Used to convert a single multiple-vectors file into multiple single-vector files.
	Single-Vector Files Conversion	Used to convert multiple single-vector files into a single multiple-vectors file.
	Matrix File Conversion	Used to convert a matrix into a list of linear combinations.
	Linear Combination File Combination	Used to convert a list of linear combinations into a matrix.
Combina-torics	Single-Element Unions	Used to compute single-element unions of collections of sets.
	Strict Single-Element Unions	Used to compute strict single-element unions of collections of sets.
Statistics	Correlation Estimation	Used to compute a variety of correlation coefficients using *R*.
	Kendall's Tau Correlation	Used to compute Kendall's tau correlation statistics.
	Mann-Whitney U Test	Used to compute Mann-Whitney U statistics.
General Utilities	Interval Comparison	Used to compare intervals for equality within a given tolerance.
	Numerical Vector Comparison	Used to compare numerical vectors for equality within a given tolerance.
	Variable Participation	Used to group mathematical expressions based on the variables they contain.

the names (or IDs) of all chemical reactions contained in *R*; and *R* can also be supplied to the *Random Constraint Generator* process to create a text file *C* of randomly generated flux constraints. The files *R*, *N*, and *C* can then be supplied to the *FBA Constraint Variation-Objective Function Analysis* process to determine the maximum fluxes of the reactions in *R* that are denoted in *N* for each set of flux constraints in *C*. Each of these files can be edited by the user at any step, and many other combinations of processes are possible.

The use of the SBRT as an application requires no programming ability, and is fully documented in a freely available HTML-based *User's Guide*, which provides a detailed description of each process and contains hyperlinks to at least one complete example. An example of the *Path Identification* process is illustrated in Figure 1.

Support for external software

The Systems Biology Research Tool's API is designed to support multiple forms of *external* software (software not included in the SBRT's API), making the SBRT highly modular and thus evolvable. A *process plug-in* is an external software package that can be written by any skilled programmer, executed as a process by the SBRT application, and shared among other users. As a consequence of the existing capabilities of the SBRT, development of process plug-ins is considerably easier and faster than development of new stand-alone applications. Plug-ins can, for example, call high-level methods from the API that perform file parsing, process monitoring, algorithm execution, and error-detection. Plug-ins can also call low-level methods to facilitate the development of novel high-level methods. Instructions for writing process plug-ins are included in the *Developer's Guide*, and an

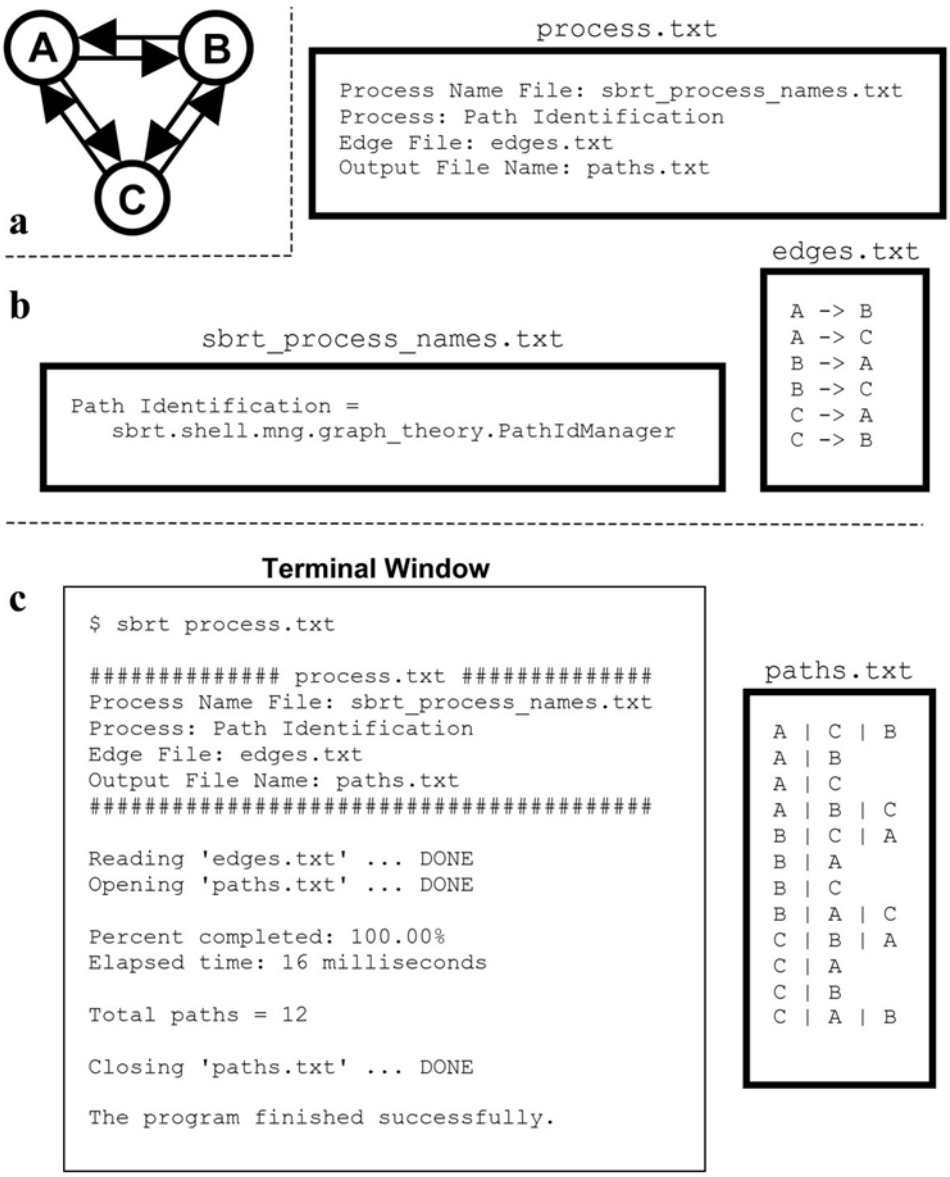


Figure 1. Identifying the simple paths in a directed graph. (**a**) The graph under consideration. (**b**) The input files to the SBRT. (**c**) The execution of the SBRT from the command line and its subsequent output. Rectangles with thick borders represent text files, with their name denoted directly above. The file `edges.txt` is created by the user to store the edges of the graph in **a**. The file `sbrt_process_names.txt` is used to define a name for the process and also provides part of the mechanism for incorporating process plug-ins. The file `process.txt` is used to organize the input, and all simple paths in the graph are identified with the command **`sbrt process.txt`**. The file `paths.txt` is created by the SBRT with a single path on each line, with nodes delimited by the pipe character.

example plug-in is also included with the package. Additionally, the SBRT's API supports communication with other forms of external software, such as applications and software libraries. The ability to interact with Mathematica, R, GLPK, CPLEX, Xerces, and Metatool [4, 5] is already implemented.

Similar software

Due to its ability to communicate with other software, the Systems Biology Research Tool provides some functionality similar to that of Cytoscape [6], CellDesigner [7], and the Systems Biology Workbench [8]. Both Cytoscape and CellDesigner can also be extended via plug-ins, but their current capabilities are substantially different from those of the SBRT. The Systems Biology Workbench is primarily intended to unify other applications by acting as a broker. The SBRT can be used in a similar way, but this is not its primary function. The SBRT can be used independently of other applications, and it also provides implementations of algorithms not currently available in any other software package [9].

Presently, the majority of processes offered by the Systems Biology Research Tool are for analyzing stoichiometric networks. Software already exists that is capable of particular types of such analysis, such as the COBRA Toolbox [10], CellNetAnalyzer [11], Metatool [4, 5], FBA3, moma [12], PathwayAnalyser [13], expa [14], YANA [15], and SNA [16]. Some of these programs are stand-alone applications (Metatool 4.x, FBA3, moma, PathwayAnalyser, expa, YANA), and the remainder can only function within a specific programming environment, such as MATLAB or Mathematica (Metatool 5.0, COBRA Toolbox, CellNetAnalyzer, SNA). In Table 2 and the following section, we compare and

Table 2. Features of the Systems Biology Research Tool and similar software packages.

Software Package	Systems Biology Research Tool	COBRA Toolbox 1.3.3	CellNet-Analyzer 9.0	Metatool 4.9.2	Metatool 5.0	Pathway-Analyser 1.0	expa	YANA 0.9.8	SNA
Provides a graphical installation procedure	✓								
Requires separate installation of other software packages for basic functionality		✓	✓		✓	✓		✓	✓
Requires commercial software for basic functionality		✓	✓						✓
Windows compatible	✓	✓	✓	✓	✓		✓	✓	
Mac compatible	✓	✓	✓		✓		✓	?	
Linux compatible	✓	✓	✓	✓	✓	✓	✓	✓	✓
Requires programming ability to use		✓			✓				✓
Can be used via a command line interface	✓	✓	✓	✓	✓	✓	✓		✓
Provides a graphical user interface	✓		✓					✓	
Provides a documented API	✓	✓	✓		In Pro-gress				

contrast some of the features and designs of these programs with that of the Systems Biology Research Tool.

Evolvability

Due to its API and support for external software, the SBRT has the ability to evolve in conjunction with the field of systems biology itself. In contrast, none of the stand-alone applications for stoichiometric network analysis listed above (Metatool 4.x, FBA3, moma, PathwayAnalyser, expa, YANA) provide both a documented API and a mechanism for the inclusion of additional software (other than by modifying existing source code). Therefore, the ability of independent software developers to expand upon these programs is greatly hindered. This is not the case, however, for software written for MATLAB or Mathematica. These mathematical programming environments both provide a large number of powerful functions, well documented API's, and mechanisms for the inclusion of external software, making the development of new software straightforward. MATLAB and Mathematica, however, are both closed-source. Consequently, certain aspects of their performance and functionality are impossible to alter, which results in additional constraints during software development and limitations during performance optimization.

Cost

To our knowledge, all of the stoichiometric network analysis software listed above is free of charge, at least for academic purposes. MATLAB and Mathematica, however, are both commercial software packages. In contrast, the SBRT is completely free of charge for every user.

Ease of use

One of the most important aspects of any software package is its ease of installation and use. The SBRT differs from the programs listed above in several ways. First, some of these programs require the installation of libraries or other programs before they can be used, while SBRT installation is self-contained and guided with a graphical user interface. Second, some of the existing programs must be used from a command line interface, which is cumbersome for the "typical" Windows user. The SBRT can be used from both the command line and from a simple graphical user interface. Third, while some existing programs require programming ability, the SBRT does not, when used as an application.

Scope

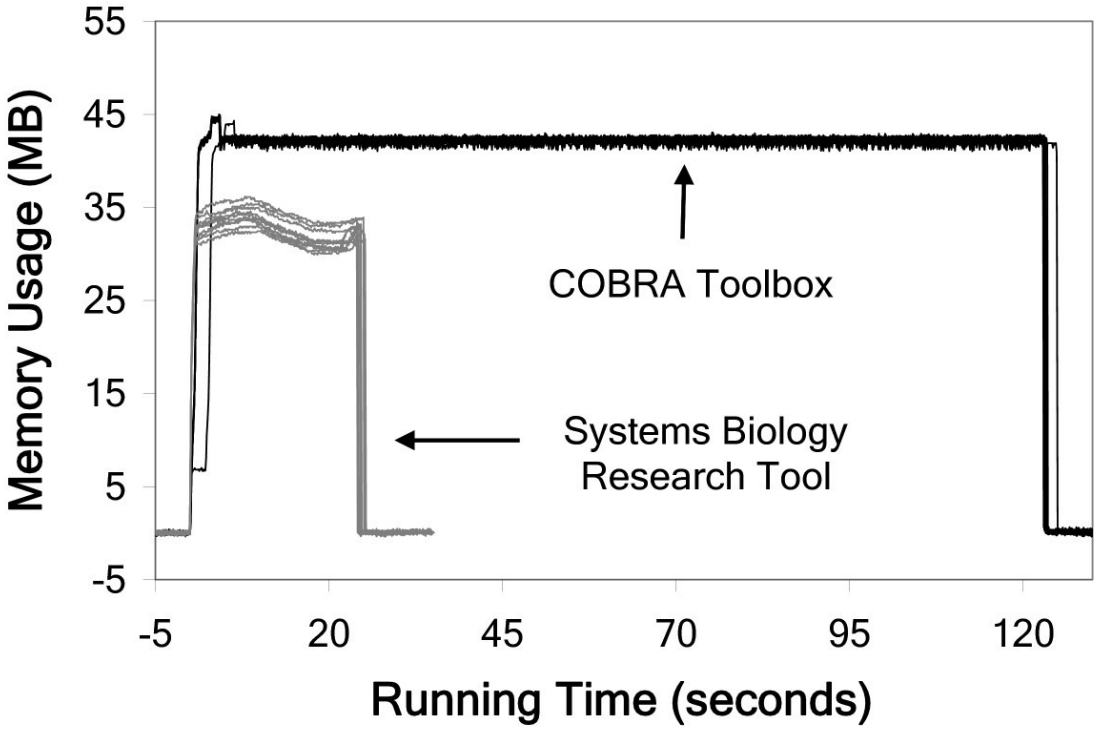
The programs listed above are intended primarily for different types of stoichiometric network analyses, and they are sometimes quite limited in scope. The SBRT, however, has been explicitly designed to integrate techniques from all of systems biology.

Performance

Of all existing packages, the COBRA Toolbox is most similar to the SBRT in terms of the computational procedures offered by both. Because of these similarities, we performed a comparative performance analysis of some capabilities offered by both packages. Specifically, we carried out 5 analyses using an *in silico* model of *S. cerevisiae* metabolism [17]. For analyses *A* and *B*, the model was provided a minimal growth-supporting medium, where the variability of all reaction rates (*A*) and the effect of all single-gene deletions on the maximum growth rate (*B*) were computed. For analyses *C*, *D*, and *E*, the model was sequentially provided 100 randomly generated growth-supporting media, where the

maximum growth rate (*C*), the variability of all reaction rates (*D*), and the effect of all single-gene deletions (*E*) were computed. The average maximum memory usage of the COBRA Toolbox was 1.30 (*A*), 1.00 (*B*), 1.01 (*C*), 0.96 (*D*), and 0.65 (*E*) times that of the SBRT; and the SBRT was 5.00 (*A*), 2.75 (*B*), 1.06 (*C*), 4.87 (*D*), and 3.73 (*E*) times faster than the COBRA Toolbox (Figure 2).

Figure 2. Memory usage vs. running time for the SBRT (grey) and COBRA Toolbox (black) for 10 executions each of analysis *A*.



Conclusions

The Systems Biology Research Tool is a technological advance for systems biology. This software can be used to make sophisticated computational techniques available to everyone, to facilitate cooperation among researchers, and to expedite progress in the field of systems biology.

Acknowledgements

We sincerely thank Christopher Lewis for his invaluable advice during software development. We also thank Christa Deiwiks, Mark Fleharty, João Rodrigues, and Annette Evangelisti for helpful discussions during this project. AW acknowledges support through SNF grant 315200-116814.

References

1. N.D. Price, J.A. Papin, C.H. Schilling, and B.Ø. Palsson. Genome-scale microbial *in silico* models: the constraints-based approach. *Trends in Biotechnology*, 21(4):162–169, 2003.
2. N.D. Price, J.L. Reed, and B.Ø. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897, 2004.
3. The Systems Biology Research Tool's homepage. http://www.bioc.uzh.ch/wagner/software/SBRT.
4. A. Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22(15):1930, 2006.
5. T. Pfeiffer, I. Sanchez-Valdenebro, J.C. Nuno, F. Montero, and S. Schuster. METATOOL: for studying metabolic networks. *Bioinformatics*, 15(3):251, 1999.
6. P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498, 2003.
7. A. Funahashi, M. Morohashi, H. Kitano, and N. Tanimura. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *Biosilico*, 1(5):159–162, 2003.
8. H.M. Sauro, M. Hucka, A. Finney, C. Wellock, H. Bolouri, J. Doyle, and H. Kitano. Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. *Omics: A Journal of Integrative Biology*, 7(4):355–372, 2003.
9. J. Wright and A. Wagner. Exhaustive identification of steady state cycles in large stoichiometric networks. *BMC Systems Biology*, 2(1):61, 2008.
10. S.A. Becker, A.M. Feist, M.L. Mo, G. Hannum, B.Ø. Palsson, and M.J. Herrgård. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox. *Nature Protocols*, 2(3):727–738, 2007.
11. S. Klamt, J. Saez-Rodriguez, and E.D. Gilles. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Systems Biology*, 1(1):2, 2007.
12. D. Segre, D. Vitkup, and G.M. Church. Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(23):15112, 2002.
13. K. Raman and N. Chandra. PathwayAnalyser: a systems biology tool for flux analysis of metabolic pathways. *Nature Precedings*, 2008.
14. S.L. Bell and B.Ø. Palsson. Expa: a program for calculating extreme pathways in biochemical reaction networks. *Bioinformatics*, 21(8):1739, 2005.
15. R. Schwarz, P. Musch, A. Von Kamp, B. Engels, H. Schirmer, S. Schuster, and T. Dandekar. YANA – a software tool for analyzing flux modes, gene-expression and enzyme activities. *BMC Bioinformatics*, 6(1):135, 2005.
16. R. Urbanczik. SNA – a toolbox for the stoichiometric analysis of metabolic networks. *BMC Bioinformatics*, 7(1):129, 2006.
17. N.C. Duarte, M.J. Herrgård, and B.Ø. Palsson. Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. *Genome Research*, 14(7):1298, 2004.

Article 3

Exhaustive identification of steady state cycles in large stoichiometric networks

Originally published as: J. Wright and A. Wagner. Exhaustive identification of steady state cycles in large stoichiometric networks. *BMC Systems Biology*, 2(1):61, 2008.

Abstract

Background: Identifying cyclic pathways in chemical reaction networks is useful, because such cycles may indicate *in silico* violation of energy conservation, or the existence of feedback *in vivo*. Our ability to identify cycles in stoichiometric networks, such as signal transduction and genome-scale metabolic networks, has been hampered by the computational complexity of the methods currently used.

Results: We describe a new algorithm for the identification of cycles in stoichiometric networks, and we compare its performance to two others by exhaustively identifying the cycles contained in the genome-scale metabolic networks of *H. pylori*, *M. barkeri*, *E. coli*, and *S. cerevisiae*. Our algorithm can substantially decrease both the execution time and maximum memory usage in comparison to the two previous algorithms.

Conclusions: The algorithm we describe improves our ability to study large, real-world, biochemical reaction networks, although additional methodological improvements are desirable.

Background

Flux balance analysis is becoming a well developed and frequently used theoretical tool to study the capabilities of large stoichiometric networks [1]. Flux balance analysis relies on several assumptions that essentially impose constraints on the allowable states of a network. Ideally, these constraints are derived from fundamental physical and chemical principles so that the physically realistic states of a network can be accurately identified and the unrealistic states ignored. Ensuring the conservation of mass can be achieved in a relatively straightforward manner, but it is much more challenging to incorporate the conservation of energy [2-8]. Methods that rely on the identification of steady state reaction cycles have been developed to achieve this [3-5]. Knowledge of these cycles can be used to constrain the direction or magnitude of certain fluxes to prevent the occurrence of thermodynamically inconsistent network states.

The incorporation of energetic constraints in flux balance analysis is not the only motivation to identify cycles. Cycles may also point towards important aspects of network function. For instance, it has been proposed that cycles in metabolic networks can affect the sensitivity and robustness of network function and allow for regulation of biochemical pathways [9]. In signal transduction networks, which are also stoichiometric in nature [10, 11], cycles may allow for feedback. Feedback is known to be an important property of signal transduction, and it has been shown to result in a variety of complex and potentially useful biochemical behaviors [12, 13].

Two algorithms have been explicitly described for the identification of steady state cycles in stoichiometric networks. Schilling, Letscher, and Palsson (SLP) described the first of

these in 2000 [14]. The SLP algorithm first defines a network of *internal* and *exchange* reactions. Internal reactions are those that are actually being studied. For metabolic networks, for example, the internal reactions are those that occur in or around the cell, such as reactions involved in glycolysis, respiration, transport of metabolites across cellular membranes, etc. Exchange reactions are pseudo-reactions that are used to supply (remove) chemical species to (from) the reaction system. The flux, or rate, of each internal reaction is constrained to be positive, while the fluxes of exchange reactions may be left unconstrained. If reversible reactions are present in the network, they are broken apart into a pair of unidirectional forward-reverse reactions, each with its own flux. These reactions are used to construct a stoichiometry matrix S which is used to formulate the equation $Sv = 0$, where v is a vector of fluxes of all reactions in the network. The solutions to this equation represent the allowable steady states of the network, where *steady* refers to the fact that the concentrations of internal chemical species remain constant.

The SLP algorithm then identifies all of the *extreme pathways* of a network, which are a unique set of flux vectors whose superposition can generate all steady-state fluxes in a network that do not violate the principle of mass conservation. The extreme pathways are then categorized based on the types of active reactions they contain. We use the word *cycles* from this point forward to refer exclusively to internal cycles, that is *type III* extreme pathways, which are the extreme pathways that do not contain active exchange reactions [14]. Type III extreme pathways are also elementary flux modes [15, 16] and extremal currents [17]. The identification of extreme pathways is equivalent to computing the set of extreme rays of a convex cone [14], which is known to be an NP-hard problem [16, 18, 19]. This computational complexity limits the size of networks for which the SLP algorithm can

be used [20], although numerous algorithmic improvements have been recently made in an effort to alleviate this problem [19, 21-23].

Mahadevan and Schilling (MS) very briefly described the second algorithm for cycle identification in 2003 [24]. In this approach, the network and its corresponding stoichiometry matrix are defined in the same way as for the SLP algorithm. The MS algorithm, however, uses a unique property of cycles to assist with their detection. If a network does not contain exchange reactions, and the fluxes of all internal reactions are constrained to the interval [0, ∞), the only reactions in the network capable of functioning will be those that participate in cycles (see Results and discussion). The MS algorithm takes advantage of this fact by using linear programming to determine the maximum flux of each reaction in the network. All of the reactions with unbounded fluxes are then used to create a sub-network of the original network. Every reaction in this sub-network necessarily participates in a cycle of the original network (see Results and discussion). The SLP algorithm is then applied to the sub-network to identify all of its extreme pathways, which are necessarily the complete set of cycles in the original network (see Results and discussion). Since the sub-network supplied to the SLP algorithm is potentially smaller than the original network, the identification of cycles in larger networks may become possible.

In this paper, we describe a new algorithm, which we refer to here as the WW algorithm. The WW algorithm is an extension of the MS algorithm, and it can reduce the size of the sub-network supplied to the SLP algorithm far beyond that of the MS algorithm. We also measure and compare, for the first time, the performances of all three algorithms using five genome-scale metabolic networks.

Materials and methods

Stoichiometric networks

Genome-scale stoichiometry matrices for *H. pylori* [28], *M. barkeri* [29], *E. coli* [30], *S. cerevisiae* [31], and *H. sapiens* [32] were constructed from the files `Hpylori_341_-model_smbl.zip`, `Mb_iAF692.xml`, `E_coli_AF1260.xml`, `Sc_iND750.xml`, and `H_sapien_Recon_1.xml`, respectively, which we obtained from [33] and [34]. We followed the SLP convention that reversible reactions are broken into forward and backward reactions, which are represented as two distinct columns in the stoichiometry matrix. Exchange reactions were not included in stoichiometry matrices. The matrices thus constructed were used to define the equation $Sv = 0$, where S denotes a stoichiometry matrix and v denotes a vector of fluxes.

Implementations

The Systems Biology Research Tool [35] (version 1.3.0) was used to create the sub-networks of the WW and MS algorithms, using the GNU Linear Programming Kit to solve all linear programs for both algorithms. Metatool [36, 37] (version 5.0), in combination with MATLAB (version 7.2), was used to execute the SLP algorithm, since it utilizes the most recent techniques for identifying elementary flux modes [22, 36]. Metatool currently uses a 32-bit binary file to identify elementary flux modes, resulting in an upper memory limit of 2^{32} bytes, that is, 4 GB.

Performance measurements

The time and memory requirements of each algorithm were used as measures of algorithmic performance. All performance measurements were made on a Dell Precision 490 computer equipped with 8 GB of RAM and a 2.33 GHz Intel Xeon processor with Kubuntu 7.10 (AMD64) as the operating system. A bash script was used to execute 10 programs sequentially for each algorithm and network. The time was recorded before each program began and after each algorithm finished execution to determine the total running time. A perl script (`memmon`) was used to frequently sample the contents of `/proc/meminfo` to monitor the memory usage during each program execution. Memory monitoring began before each algorithm was executed to establish a baseline, and the maximum memory consumption during algorithm execution was measured relative to this baseline.

Results and discussion

The following is a list of properties of extreme pathways that have been published previously [14, 16], and a list of logical deductions we make based on these properties. Each of these properties and deductions is very simple, but taken together, they lead to the principle (Deduction 8) that allows the WW algorithm to be much more efficient than its predecessors.

Property 1. Extreme pathways are composed of the minimum number of reactions needed to function (that is, to have non-zero flux) at steady state [16].

Property 2. Extreme pathways are systematically independent, that is, extreme pathways cannot be composed of other extreme pathways [14, 16].

Property 3. Type III extreme pathways contain only internal reactions, that is, they never contain exchange reactions. All other types of extreme pathways contain at least one exchange reaction [14].

Deduction 1. Given Property 1, if any reaction is removed from an extreme pathway, or inactivated by constraining its flux to zero, the entire pathway is rendered inactive, that is, the flux of each reaction in the pathway must be zero (assuming those reactions do not participate in other, active extreme pathways).

Deduction 2. Given Property 1, an extreme pathway can be viewed as a single, independent functional unit. For example, if all reactions in a network are inactivated, except those participating in a given extreme pathway, that particular pathway can still have non-zero flux.

Deduction 3. Given Property 1, a single unidirectional internal reaction can never be an extreme pathway, because it cannot function at steady state by itself (excluding 'null' reactions, like A → A).

Deduction 4. A pair of unidirectional forward-reverse reactions can form a type III extreme pathway. The pair can function at steady state without the need for other reactions and only by functioning together, satisfying Property 1. The pair cannot be composed of other extreme pathways, since a single reaction can never be an extreme pathway (Deduction 3), satisfying Property 2.

Deduction 5. If an extreme pathway is composed of only two unidirectional reactions, they must be a forward-reverse reaction pair (R_F, R_R). The only way a reaction pair can maintain

steady state concentrations is if R_R consumes the products of R_F at the same rate that R_F consumes the products of R_R.

Deduction 6. Given Properties 1 and 2, and Deduction 4, an extreme pathway composed of more than two reactions can never contain a forward-reverse reaction pair.

Deduction 7. Given Property 3 and Deduction 2, if the exchange reactions in a stoichiometric network are removed, only type III extreme pathways will remain. Note, however, that the set of type III pathways in a network may change as a result of exchange reaction removal.

Deduction 8. If a unidirectional internal reaction R is active (i.e. its flux is greater than zero), all reverse reactions of R are inactivated (i.e. their fluxes have been constrained to zero), and the network contains no exchange reactions, R necessarily participates in a type III extreme pathway composed of more than two reactions. This statement is a direct consequence of Deductions 1, 4, 6 and 7.

The WW algorithm

Step 1. Ensure that the network does not contain exchange reactions, and constrain the flux of each internal reaction to the interval [0, ∞).

Step 2. Let *U* denote an empty set, and for each reaction *R* in the network, do the following:

Step i. By pair-wise comparison, identify all reactions in the network that are the reverse of *R*.

Step ii. For each of the reverse reactions identified in Step i, constrain its flux to be zero.

Step iii. Determine if the flux of reaction *R* can assume a value other than zero, while still satisfying all currently defined constraints. If so, add reaction *R* to the set *U*.

Step iv. For each of the reverse reactions identified in Step i, restore the constraint on its flux to [0, ∞).

Step 3. Identify the extreme pathways in the network defined by the reactions contained in the set *U*.

Step 4. By pair-wise comparison, identify each pair of internal forward-reverse reactions, R_F and R_R, in the stoichiometric network. A particular R_F-R_R pair forms a single type III extreme pathway.

Step 5. Combine the results of Steps 3 and 4 to obtain the set of all cycles present in the network.

Step 1 ensures that the only extreme pathways present in the network are of type III, since all other types of extreme pathways must contain active exchange reactions. Step 2 is used to identify all reactions that participate in cycles composed of three or more reactions. Under these circumstances, the fluxes of reactions that participate in such cycles can assume any value on the interval [0, ∞), and the fluxes of reactions that do *not* participate in such cycles must be zero. Any technique capable of making this distinction, such as linear programming, can be used during this step. The reason for this property is that if a reaction *R* only participates in cycles composed of two reactions, it must be inactive when its reverse reactions are constrained to be inactive. In Step 3, a sub-network is created that is composed of the reactions identified in Step 2, and an algorithm is applied to that network

to identify the type III extreme pathways it contains. That algorithm could be the SLP algorithm, or another that achieves the same result. Step 4 is used to identify all of the extreme pathways composed of only two reactions. This could also be accomplished during Step 2.i, but it is listed separately here for the sake of clarity.

There are two key differences between the WW and MS algorithms. Firstly, the MS algorithm, as it is defined, individually maximizes the flux of each reaction in a network. Flux maximization can also be performed to accomplish Step 2.iii of the WW algorithm, but other (potentially faster) techniques could be used as well. Secondly, the MS algorithm creates sub-networks composed of all of the reactions in a network that participate in cycles. The WW algorithm, however, creates sub-networks composed only of those reactions that participate in cycles composed of three or more reactions. Consequently, the WW algorithm can sometimes produce sub-networks drastically smaller than those of the MS algorithm, which is demonstrated below.

Performance comparisons

We tested the performance of the MS, SLP, and WW algorithms for genome-scale metabolic networks from four different microbes and from *Homo sapiens* (human), comprising between 486 and 2786 chemical species, and between 642 and 4482 reactions. Our results show that both execution time efficiency (Figure 1 and 4) and memory efficiency (Figures 2, 3, 5 and 6) were substantially different for the three algorithms, with the WW algorithm being more efficient than the MS algorithm, which was more efficient than the SLP algorithm. These performance differences are described in detail in the following sections.

Preprocessing

Both the WW and MS algorithms use a preprocessing procedure to reduce the size of the network before identification of cycles is attempted. For both algorithms, the time and memory consumption of the preprocessing phase increases as network size increases, although not substantially; and WW preprocessing consumes more time and moderately more memory than MS preprocessing (Figures 1, 2, and 5).

Extreme pathway identification

The time and memory consumption of the MS, SLP, and WW algorithms during extreme pathway identification were quite different. To begin with, the SLP algorithm was unable to complete execution with the *E. coli*, *S. cerevisiae*, and *H. sapiens* networks due to memory exhaustion. This also occurred for both the MS and WW algorithms with the *H. sapiens* network. Therefore, at least for the hardware and software configurations used for these analyses, memory efficiency during extreme pathway identification was the most important factor for achieving successful computation. For the microbial networks, the maximum memory consumption of the MS algorithm was much less than that of the SLP algorithm, and the memory consumption of the WW algorithm was less than that of the MS algorithm (Figure 3 and 6). The memory usage of both the MS and SLP algorithms increased substantially with increasing network size, but in contrast, the memory usage of the WW algorithm changed little as the size of the microbial networks increased (Figure 3 and 6). Similar trends were observed for time efficiency, as well (Figure 1 and 4).

Figure 1. The average execution time of each algorithm with each network. Light grey indicates the time spent in the preprocessing phase, and dark grey indicates the time spent identifying extreme pathways. A dagger (†) directly over a bar indicates that the algorithm halted due to memory exhaustion. The indicated values are, therefore, only a lower bound on the total execution time required for successful completion.

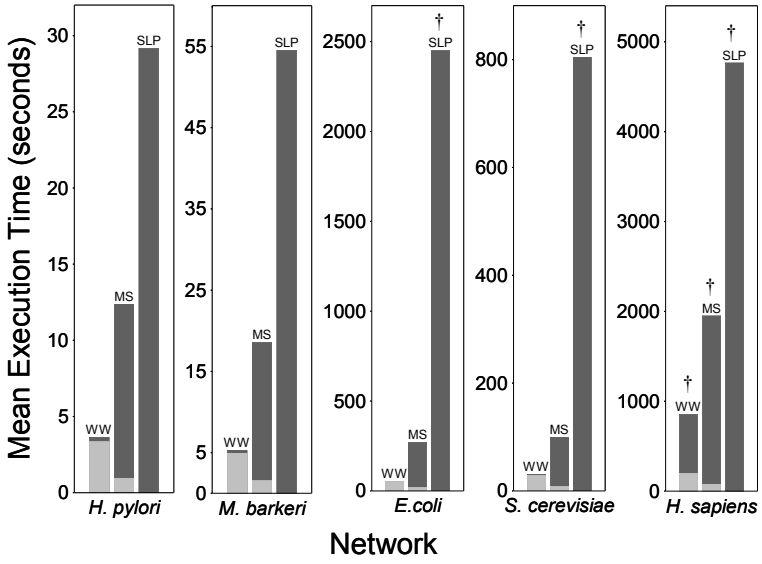


Figure 2. The average maximum memory consumption of the preprocessing phase of the MS and WW algorithms with each network.

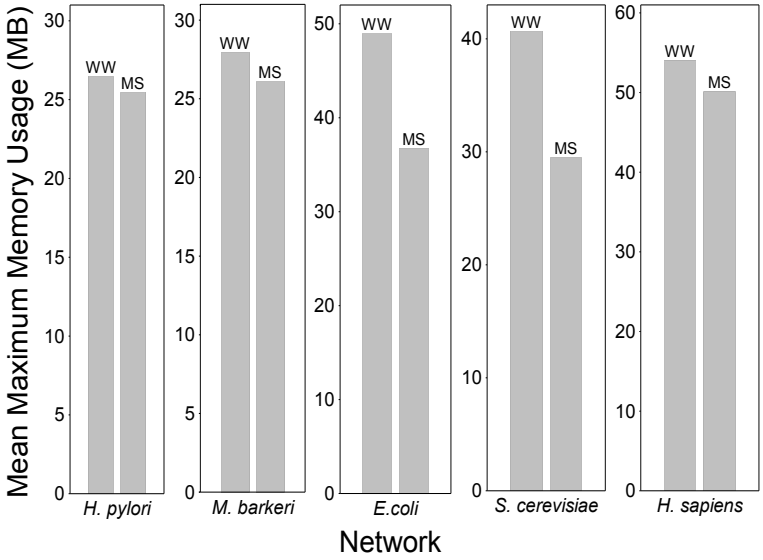


Figure 3. The average maximum memory consumption of the extreme pathway identification phase of each algorithm with each network. A dagger (†) directly over a bar indicates that the algorithm halted due to memory exhaustion. The indicated values are, therefore, only a lower bound on the maximum memory required for successful completion.

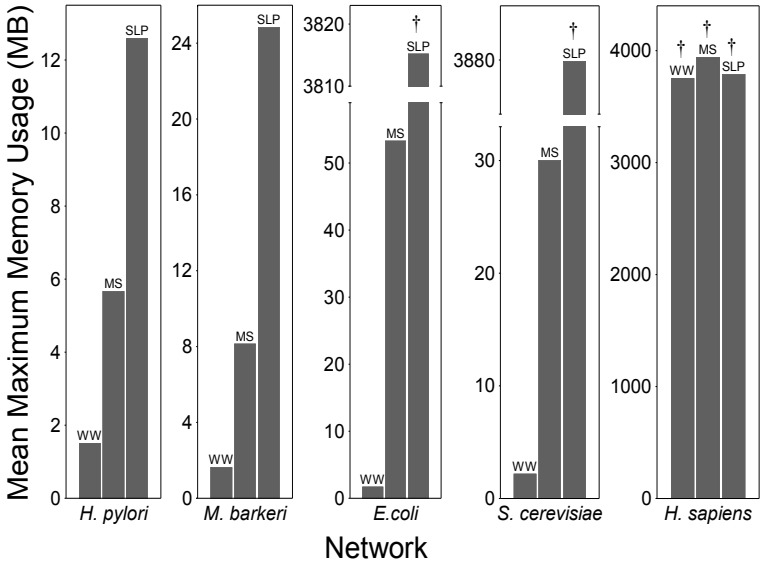


Figure 4. The average execution time of each algorithm with each network. A dagger (†) directly over a bar indicates that the algorithm halted due to memory exhaustion. The indicated values are, therefore, only a lower bound on the total execution time required for successful completion.

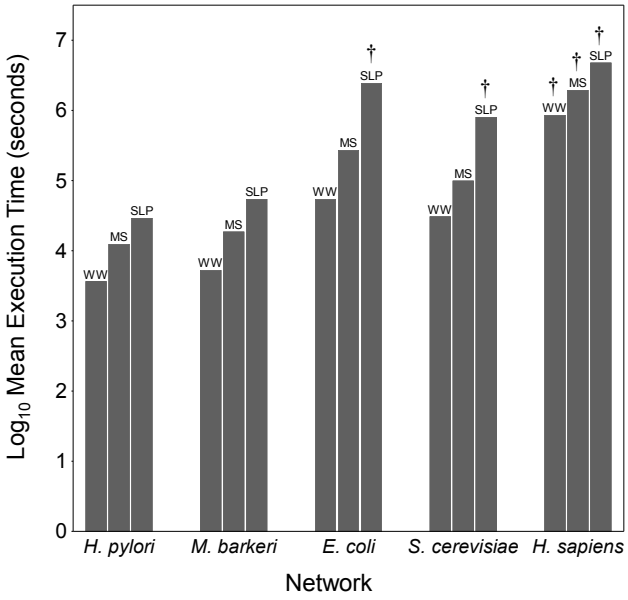


Figure 5. The average maximum memory consumption of the preprocessing phase of the MS and WW algorithms with each network.

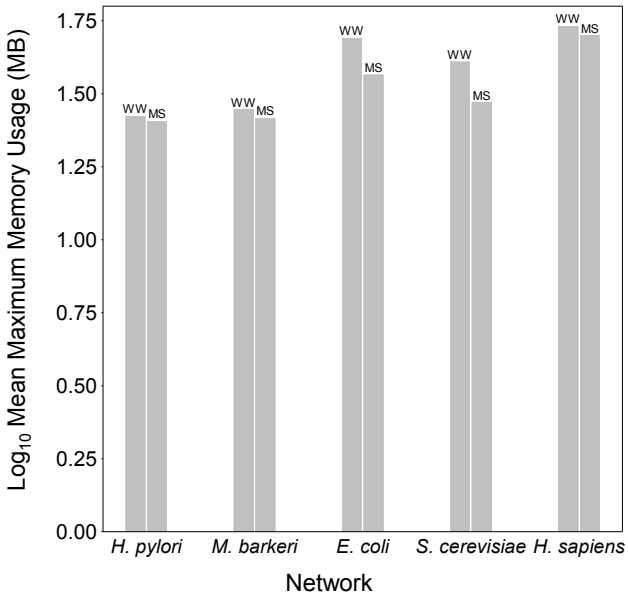


Figure 6. The average maximum memory consumption of the extreme pathway identification phase of each algorithm with each network. A dagger (†) directly over a bar indicates that the algorithm halted due to memory exhaustion. The indicated values are, therefore, only a lower bound on the maximum memory required for successful completion.

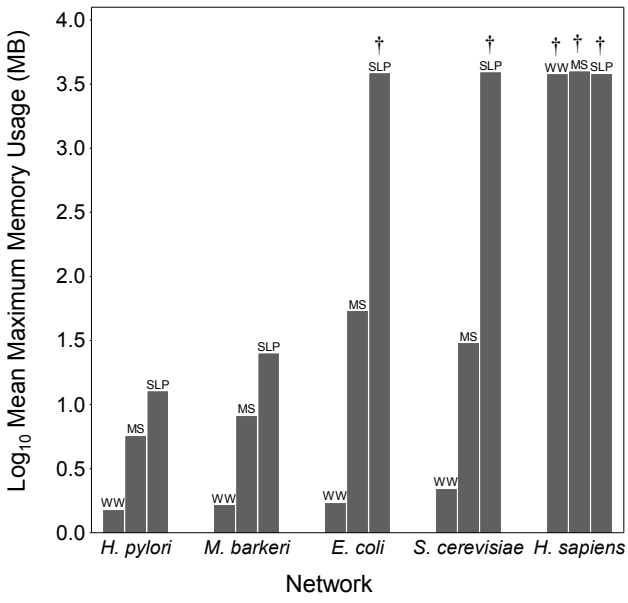


Table 1. Sizes of networks supplied to the SLP algorithm by each algorithm.

	Chemical Species			Reactions			Non-Zero Matrix Elements		
Network	**SLP**	**MS**	**WW**	**SLP**	**MS**	**WW**	**SLP**	**MS**	**WW**
H. pylori	486	251	34	642	335	26	2,861	1,285	110
M. barkeri	628	308	34	816	400	40	3,781	1,609	182
E. coli	1,673	897	57	2,635	1,140	50	10,487	3,366	201
S. cerevisiae	1,061	636	77	1,580	873	98	6,746	3,182	384
H. sapiens	2,786	1,630	633	4,482	2,812	1229	17,571	9,677	5,010

The observed performance differences of extreme pathway identification can best be explained by noting the reduction of network sizes achieved during the preprocessing phases of the MS and WW algorithms. The SLP algorithm uses the entire network, whose numbers of reactions and metabolites are shown in Table 1. The MS algorithm uses sub-networks containing only reactions that participate in cycles, but the WW algorithm uses sub-networks containing only reactions that participate in cycles that are composed of more than two reactions. The reduction in size between the full network and the sub-network used by the WW algorithm is dramatic, spanning more than an order of magnitude for each of the microbial networks (Table 1). For these networks, the majority of cycles are composed of only two reactions (Table 2), allowing the WW algorithm to produce much smaller sub-networks than the MS algorithm. This greatly reduces the computational effort required of the SLP algorithm, resulting in better performance.

Overall performance

Although the WW preprocessing step consumes more time and memory than MS preprocessing, the time and memory efficiency of identifying extreme pathways during the WW algorithm is dramatically better. Consequently, the overall execution time of the WW algorithm is substantially less than the MS algorithm, and the memory consumption is also

Table 2. Number of cycles per microbial network.

Network	Cycles of Size = 2	Cycles of Size > 2	Total Cycles
H. pylori	165	7	172
M. barkeri	200	27	227
E. coli	564	27	591
S. cerevisiae	434	39	473

significantly reduced during the phase of the computation that is most memory-intensive and most susceptible to combinatorial explosion.

Caveats

There are certain extreme cases where these performance differences will not persist. For example, if every reaction in a stoichiometric network participates in a cycle, the MS algorithm will fail to outperform the SLP algorithm. In this situation, the maximum flux of each reaction will be unbounded, resulting in a "sub-network" that is exactly the same as the original. Similarly, if every reaction in a stoichiometric network participates in cycles composed of more than two reactions, neither the MS nor WW algorithm will outperform the SLP algorithm. If all the cycles in a stoichiometric network are composed of more than two reactions, the WW algorithm will fail to outperform the MS algorithm. If a reaction network does not contain any cycles, the MS algorithm will likely have a performance advantage over the WW algorithm, depending on implementation details. We note that these extreme scenarios are not realistic for genome-scale metabolic networks (Table 2), the kinds of networks for which application of the WW algorithm would be most fruitful. If, finally, a network only contains cycles composed of two reactions, the WW algorithm will never make use of the SLP algorithm, which eliminates the chance of combinatorial explosion and most likely provides further dramatic performance improvements over the MS algorithm.

It should also be noted that the algorithmic performances described herein are dependent upon implementation details and the choice of underlying algorithms. Other software and algorithms [25-27], for example, could be used to identify extreme pathways, which would certainly change the time and memory requirements of these computations. Additionally, flux maximization was performed by both implementations of the WW and MS algorithms during the preprocessing phase. As mentioned above, replacing flux maximization with another technique, such as an infeasibility test, would also change the time and memory requirements of this portion of the algorithms.

Conclusions

The WW algorithm consistently achieves significant performance improvements over both the MS and SLP algorithms for the microbial networks we examined. For these networks, the execution time and maximum memory consumption of the WW algorithm are both smaller by multiple factors. The scaling behavior of the WW algorithm as a function of network size is also preferable to both the MS and SLP algorithms. Due to combinatorial explosion during extreme pathway identification, however, all of the algorithms fail to identify the cycles within the human metabolic network. At this point in time, the WW algorithm appears to be the best choice for identifying steady state cycles in large, real-world stoichiometric networks, although additional algorithmic innovation is clearly desirable.

Acknowledgements

We thank Christopher Lewis for his invaluable advice during software development, Axel von Kamp for his helpful discussions concerning Metatool, and João Rodrigues for his suggestions during preparation of this manuscript. AW acknowledges support from Swiss National Science Foundation (315200-116814), the Swiss Initiative in Systems Biology (SystemsX), and the Santa Fe Institute.

References

1. N.D. Price, J.L. Reed, and B.Ø. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897, 2004.
2. D.A. Beard, S. Liang, and H. Qian. Energy balance for analysis of complex metabolic networks. *Biophysical Journal*, 83(1):79–86, 2002.
3. N.D. Price, I. Famili, D.A. Beard, and B.Ø. Palsson. Extreme pathways and Kirchhoff's second law. *Biophysical Journal*, 83(5):2879–2882, 2002.
4. N.D. Price, I. Thiele, and B.Ø. Palsson. Candidate states of *Helicobacter pylori*'s genome-scale metabolic network upon application of "loop law" thermodynamic constraints. *Biophysical Journal*, 90(11):3919–3928, 2006.
5. A. Kümmel, S. Panke, and M. Heinemann. Systematic assignment of thermodynamic constraints in metabolic network models. *BMC Bioinformatics*, 7(1):512, 2006.
6. D.A. Beard, E. Babson, E. Curtis, and H. Qian. Thermodynamic constraints for biochemical networks. *Journal of Theoretical Biology*, 228(3):327–333, 2004.
7. F. Yang, H. Qian, and D.A. Beard. Ab initio prediction of thermodynamically feasible reaction directions from biochemical network stoichiometry. *Metabolic Engineering*, 7(4):251–259, 2005.
8. R. Nigam and S. Liang. A pivoting algorithm for metabolic networks in the presence of thermodynamic constraints. In *Computational Systems Bioinformatics Conference, 2005. Proceedings. 2005 IEEE*, pages 259–267. IEEE, 2005.
9. H. Qian and DA Beard. Metabolic futile cycles and their functions: a systems analysis of energy and control. *IEE Proceedings-Systems Biology*, 153(4):192–200, 2006.
10. J.A. Papin and B.Ø. Palsson. The JAK-STAT signaling network in the human B-cell: an extreme signaling pathway analysis. *Biophysical Journal*, 87(1):37–46, 2004.
11. J.A. Papin and B.Ø. Palsson. Topological analysis of mass-balanced signaling networks: a framework to obtain network properties including crosstalk. *Journal of Theoretical Biology*, 227(2):283–297, 2004.
12. J.E. Ferrell. Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability. *Current Opinion in Cell Biology*, 14(2):140–148, 2002.
13. M. Freeman. Feedback control of intercellular signalling in development. *Nature*, 408(6810):313–319, 2000.
14. C.H. Schilling, D. Letscher, and B.Ø. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, 203(3):229–248, 2000.
15. S. Schuster, D.A. Fell, and T. Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, 18(3):326–332, 2000.
16. J.A. Papin, J. Stelling, N.D. Price, S. Klamt, S. Schuster, and B.Ø. Palsson. Comparison of network-based pathway analysis methods. *Trends in Biotechnology*, 22(8):400–405, 2004.
17. C. Wagner and R. Urbanczik. The geometry of the flux cone of a metabolic network. *Biophysical Journal*, 89(6):3837–3845, 2005.

18. J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5(1):175, 2004.

19. M. Terzer and J. Stelling. Accelerating the computation of elementary modes using pattern trees. *Algorithms in Bioinformatics*, pages 333–343, 2006.

20. J.A. Papin, N.D. Price, S.J. Wiback, D.A. Fell, and B.Ø. Palsson. Metabolic pathways in the post-genome era. *Trends in Biochemical Sciences*, 28(5):250–258, 2003.

21. R. Urbanczik and C. Wagner. An improved algorithm for stoichiometric network analysis: theory and applications. *Bioinformatics*, 21(7):1203, 2005.

22. S. Klamt, J. Gagneur, and A. von Kamp. Algorithmic approaches for computing elementary modes in large biochemical reaction networks. *IEE Proceedings-Systems Biology*, 152:249, 2005.

23. S.L. Bell and B.Ø. Palsson. Expa: a program for calculating extreme pathways in biochemical reaction networks. *Bioinformatics*, 21(8):1739, 2005.

24. R. Mahadevan and C.H. Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic Engineering*, 5(4):264–276, 2003.

25. T.S. Motzkin, H. Raiffa, G.L. Thompson, and R.M. Thrall. The double description method. *Contributions to the Theory of Games*, 2:51–73, 1953.

26. D. Avis. Computational experience with the reverse search vertex enumeration algorithm. *Optimization Methods and Software*, 10(2):107–124, 1998.

27. D. Avis. A revised implementation of the reverse search vertex enumeration algorithm. In *Polytopes–Combinatorics and Computation*, volume 29, pages 177–198.

28. I. Thiele, T.D. Vo, N.D. Price, and B.Ø. Palsson. Expanded metabolic reconstruction of *Helicobacter pylori* (iIT341 GSM/GPR): an *in silico* genome-scale characterization of single-and double-deletion mutants. *Journal of Bacteriology*, 187(16):5818, 2005.

29. A.M. Feist, J.C.M. Scholten, B.Ø. Palsson, F.J. Brockman, and T. Ideker. Modeling methanogenesis with a genome-scale metabolic reconstruction of *Methanosarcina barkeri*. *Molecular Systems Biology*, 2(1), 2006.

30. A.M. Feist, C.S. Henry, J.L. Reed, M. Krummenacker, A.R. Joyce, P.D. Karp, L.J. Broadbelt, V. Hatzimanikatis, and B.Ø. Palsson. A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Molecular Systems Biology*, 3(1), 2007.

31. N.C. Duarte, M.J. Herrgård, and B.Ø. Palsson. Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. *Genome Research*, 14(7):1298, 2004.

32. N.C. Duarte, S.A. Becker, N. Jamshidi, I. Thiele, M.L. Mo, T.D. Vo, R. Srivas, and B.Ø. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proceedings of the National Academy of Sciences*, 104(6):1777, 2007.

33. Systems Biology Research Group. http://gcrg.ucsd.edu.

34. BiGG Database. http://bigg.ucsd.edu.

35. J. Wright and A. Wagner. The Systems Biology Research Tool: evolvable open-source software. *BMC Systems Biology*, 2(1):55, 2008.

36. A. Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22(15):1930, 2006.

37. T. Pfeiffer, I. Sanchez-Valdenebro, J.C. Nuno, F. Montero, and S. Schuster. METATOOL: for studying metabolic networks. *Bioinformatics*, 15(3):251, 1999.

Article 4

Batch and continuous culture-based selection strategies for acetic-acid tolerance in xylose-fermenting *Saccharomyces cerevisiae*

Originally published as: J. Wright, E. Bellissimi, E. de Hulster, A. Wagner, J.T. Pronk, and A.J.A. van Maris. Batch and continuous culture-based selection strategies for acetic-acid tolerance in xylose-fermenting *Saccharomyces cerevisiae*. *FEMS Yeast Research*, 2011.

Abstract

Acetic-acid tolerance of *Saccharomyces cerevisiae* is crucial for yeast-based production of bioethanol and other bulk chemicals from lignocellulosic plant-biomass hydrolysates. Acetic acid released during hydrolysis inhibits yeast growth and metabolism, especially at low pH. Targeted metabolic engineering is hindered by the complex, multi-factorial nature of acetic-acid tolerance. This study explores two evolutionary engineering strategies for improvement of acetic-acid tolerance of the xylose-fermenting strain *S. cerevisiae* RWB218, whose anaerobic growth on xylose at pH 4 is inhibited at acetic-acid concentrations above 1 g L^{-1}: (i) sequential anaerobic, pH-controlled batch cultivation (pH 4) at increasing acetic-acid concentrations and (ii) prolonged cultivation in anaerobic continuous cultures without pH control, in which acidification caused by ammonium assimilation generates a selective pressure for improved acetic-acid tolerance. After ca. 400 generations, the sequential-batch and continuous selection cultures grew on xylose at pH ≤ 4 in the presence of 6 g L^{-1} and 5 g L^{-1} acetic acid, respectively. In the continuous cultures, the specific xylose-consumption rate had increased by 75% to 1.7 g xylose (g biomass)$^{-1}$ h^{-1}. After storage of samples from both selection experiments at -80 ^{o}C and cultivation in the absence of acetic acid, they failed to grow on xylose at pH 4 in the presence of 5 g L^{-1} acetic acid. Characterization in chemostat cultures with linear acetic-acid feeding gradients demonstrated a strong acetate-inducible acetic-acid tolerance in samples from the continuous selection protocol. This provides a valuable platform for analysis and improvement of acetic acid tolerance and its regulation.

Introduction

Evolutionary engineering is a rational approach for obtaining microorganisms with industrially desirable phenotypes [1], based on mutation and selection. Of the frequent mutations that occur within microbial cultures, some enable the host cell to grow and reproduce more effectively. The growth environment largely determines whether and to what extent a particular mutation and its resulting phenotype are beneficial to the host and the environment is thereby said to "select" certain phenotypes. The key challenge of the evolutionary engineer is to design, test and develop cultivation strategies that effectively select cells with desirable phenotypes. For selection to occur, at least two phenotypes must be present, either from the onset of the culture or arising during cultivation. Selection of desirable microbial phenotypes can be 'artificial' (i.e. using man-made devices such as colony pickers or cell sorters [2], or 'natural', by allowing mixtures of cells with differing phenotypes to compete for common resources during cultivation. Herein, we use the term 'selection' to refer exclusively to the latter.

A brief description of two popular cultivation techniques illustrates how laboratory cultivation can be used to select for particular phenotypes. In a typical chemostat culture, a single growth-limiting nutrient is continuously present at a low concentration [3, 4]. All else being constant over time, long-term chemostat cultivation will therefore select for cells with an increased affinity, i.e. cells that can achieve a higher specific growth rate at a given suboptimal concentration of the growth-limiting nutrient. Conversely, in a typical batch culture, all nutrients are initially in excess, and nutrient limitation only occurs briefly before the culture enters stationary phase due to nutrient depletion. Consequently, selection in

batch cultures will tend to favor cells that can grow fast at non-limiting substrate concentrations. Selection strategies may be further improved by, for example, the application of dynamic feeding regimes [5] or by the application of chemical or physical stresses [6].

A combination of evolutionary engineering in batch and chemostat cultures has been applied successfully to improve the kinetics of xylose- and arabinose-fermentation by genetically engineered strains of bakers' yeast (*Saccharomyces cerevisiae*) [5, 7, 8], with the goal to enable fuel ethanol production from non-food lignocellulosic plant biomass. However, fermentation of these pentose sugars is not the only challenge for yeast-based ethanol production from such feedstocks, as several inhibitors of yeast growth and metabolism are released during hydrolysis of lignocellulose [9, 10]. A particularly important inhibitor in lignocellulosic hydrolysates is acetic acid, which is released upon hydrolysis of acetyl groups from the carbohydrate polymers present in plant biomass [11-13]. Especially at low pH, acetic acid is a strong inhibitor of microbial metabolism and growth, which explains its common use as a food preservative. Development of pentose-fermenting yeast strains with an improved tolerance to acetic acid offers an interesting approach to the seemingly unavoidable presence of acetic acid in lignocellulosic hydrolysates.

Under conditions relevant for yeast cultivation, the weak organic acid (pK=4.76) acetic acid exists in two forms – protonated and unprotonated. The protonated form is relatively non-polar, which allows it to passively diffuse across the (hydrophobic) plasma membrane. Alternatively, acetic acid can enter yeast cells via the Fps1p aquaglyceroporin [14, 15]. Independent of the mechanism of entry into the cytosol, where the pH is near-neutral, dissociation into a proton and acetate ion occurs. Intracellular accumulation of

protons and acetate anions can interfere with the function of some enzymes [16], thus causing inhibition of metabolism and growth. Many microbes, including *S. cerevisiae*, have transmembrane proteins that expel protons and organic anions from the cytosol. These generally require a net input of free energy to drive ion export, e.g. via ATP hydrolysis [16-18]. At a low extracellular pH, exported acetate and protons may reassociate and diffuse back into the cell, leading to a cyclic process in which the plasma membrane proton-motive force is dissipated. Competition of (cyclic) energy-dependent ion transport with free-energy-requiring, growth-related cellular processes is likely to contribute to growth inhibition by acetic acid.

Based on current knowledge of acetic acid inhibition, several metabolic engineering approaches might be envisaged to improve tolerance to acetic acid. For example, characteristics of the plasma membrane could be altered to decrease the rate of diffusion of acetic acid into the cytosol, diffussion facilitating proteins can be deleted [15], cytosolic proteins might be altered to tolerate higher intracellular concentrations of protons and acetate ions, or the rate of ion export could be increased by altering the number, type, or activity of proton and acetate exporters in the membrane. Furthermore, ATP availability could be increased by increasing the sugar consumption flux, or by reducing the ATP requirement of other cellular processes. Implementing such strategies is, however, extremely difficult due to our limited understanding of the complex and multifactorial nature of acetic-acid tolerance and sensitivity. This provides a strong incentive to explore the potential of evolutionary engineering for increasing acetic-acid tolerance of *S. cerevisiae* as this approach does not require *a priori* knowledge of the molecular basis of cellular tolerance.

The xylose-fermenting *S. cerevisiae* strain RWB218 used in this study was derived previously from the laboratory strain CEN.PK through a combination of metabolic engineering and evolutionary engineering [8]. As observed in other xylose-fermenting *S. cerevisiae* strains, kinetics of xylose fermentation are strongly affected by the presence of acetic acid at low pH, especially in the absence of glucose [19]. The goal of the present study was to investigate whether acetic-acid tolerance of an engineered, xylose fermenting *S. cerevisiae* strain can be increased via evolutionary engineering in two different experimental set-ups: (i) sequential anaerobic, pH controlled batch cultivation on xylose at gradually increasing concentrations of acetic acid and (ii) prolonged cultivation in anaerobic, xylose-grown and acetic-acid supplemented continuous cultures without pH control, in which acidification due to the consumption of ammonium provides a continuous selection pressure for cells with improved acetic-acid tolerance.

Materials and methods

Strains and maintenance

Saccharomyces cerevisiae RWB218 is a genetically and evolutionarily engineered xylose-utilizing strain that expresses the *Piromyces* XylA (xylose isomerase) gene and in which the enzymes of the nonoxidative pentose-phosphate pathway have been overexpressed [8]. Stock cultures were grown at 30 °C in shake flasks on a synthetic medium supplemented with 20 g L^{-1} glucose. When the stationary phase was reached, sterile glycerol was added to 30% (v/v), and 2-mL aliquots were stored in sterile vials at -80 °C. For storage of the long term selection runs, culture samples were centrifuged, resuspended in synthetic medium supplemented with 30% (v/v) sterile glycerol and stored at -80 °C for further characterization. Material transfer requests for strain RWB218 should be addressed to Royal Nedalco (info@nedalco.nl, for the attention of J.J. den Ridder).

Cultivation and media

Shake-flask cultivation was performed at 30 °C in a synthetic medium [20]. The pH of the medium was adjusted to 6.0 with 2 M KOH prior to sterilization. Precultures were prepared by inoculating 100 ml medium containing 20 g L^{-1} xylose in a 500 ml shake-flask with a frozen stock culture. After 2 to 3 days incubation at 30 °C in an orbital shaker (200 rpm), this culture was used to inoculate fermentor cultures.

All fermentations were carried out at 30 °C in 2-liter laboratory fermentors (Applikon, Schiedam, The Netherlands) with a working volume of 1 liter. The culture pH was kept at pH 4.0 by automatic addition of 2 M KOH, except for the prolonged selection in

continuous culture. Cultures were stirred at 600 rpm and sparged with 0.5 l min^{-1} nitrogen (<10 ppm oxygen). Dissolved oxygen was monitored with an autoclavable oxygen electrode (Applisens, Schiedam, The Netherlands). Synthetic medium [20] was used containing xylose as the carbon source, supplemented with 100 µl L^{-1} of silicone antifoam (Sigma, antifoam 204) as well as with anaerobic growth factors ergosterol (0.01 g L^{-1}) and Tween 80 (0.42 g L^{-1}) dissolved in ethanol [19], resulting in 11-13 mM ethanol in the medium. To minimize diffusion of oxygen, fermentors were equipped with Norprene tubing (Cole Parmer Instrument Company, Vernon Hills, USA), and the medium vessel was sparged with nitrogen gas during continuous fermentations.

For the sequential batch cultivations, the fraction of CO_2 measured in the effluent gas was used to estimate the specific growth rate of each batch, and the cumulative CO_2 production was used to automatically determine when to remove ~99.5% of the culture broth and refill the fermentor with fresh synthetic medium, which also enabled consistent quantification of batch durations.

For continuous selection (D=0.05 h^{-1}) the pH of the medium was adjusted to 4.25 with KOH, but the pH in the fermentor was not maintained at a constant value. The acetic acid concentration in the supplied medium was periodically increased from an initial concentration of 1 g L^{-1} to a final concentration of 5 g L^{-1}. During the acetic acid gradients, the specific xylose consumption rates were calculated from the xylose mass balance for which the change in the xylose concentration was estimated from the derivative of polynomal spline functions.

To obtain a smoothly increasing acetic-acid concentration gradient in continuous cultures, a gradient mixer consisting of two 20 L medium vessels containing 0 and 19 g L^{-1} of acetic acid, respectively, was connected to steady-state anaerobic xylose-limited cultures at a dilution rate of 0.05 h^{-1} and pH 4. Acetic acid supplemented medium was fed to the medium vessel lacking acetic acid at a flow rate equal to the medium supply of medium to the culture.

Determination of culture dry weight

Culture samples (10.0 or 20.0 ml) were filtered over preweighed nitrocellulose filters (pore size 0.45 µm; Gelman laboratory, Ann Arbor, USA). After removal of medium, the filters were washed with demineralized water and dried in a microwave over (Bosch, Stutgart, Germany) for 20 min at 360 W and weighed.

Gas analysis

Exhaust gas was cooled in a condensor (2 °C) and dried with a Permapure dryer type MD-110-48P-4 (Permapure, Toms River, USA). O_2 and CO_2 concentrations were determined with an NGA 2000 analyzer (Rosemount Analytical, Orrville, USA).

Metabolite analysis

The supernatant obtained following centrifugation of culture samples was analyzed for xylose, organic acids, glycerol, and ethanol via HPLC analysis on a Waters Alliance 2690 HPLC (Waters, Milford, USA) containing a Biorad HPX 87H column (Biorad, Hercules, USA). The column was eluted at 60 °C with 0.5 g L^{-1} H_2SO_4 at a flow rate of 0.6 ml min^{-1}. Detection

was performed using a Waters 2410 refractive-index detector and a Waters 2487 UV detector.

Results

Prolonged repetitive batch cultivation with increasing acetic-acid concentrations

At pH 4, anaerobic growth on xylose of *S. cerevisiae* RWB218 is already significantly inhibited at an acetic acid concentration of 1 g L^{-1}, while no growth is observed at acetic-acid concentrations above 2 g L^{-1} (data not shown). To select for cells capable of growth at higher acetic-acid concentrations, RWB218 was grown anaerobically on xylose in 54 sequential batch reactor (SBR) cultures, covering a total cultivation period of 7 months. Over this period, the concentration of acetic acid was gradually increased from 0 to 6 g L^{-1} by discrete increments of 1 g L^{-1} (Figure 1). Although the lag phase decreased during the first 6 cycles, which were grown in the absence of acetic acid, the specific growth rate on xylose remained constant at 0.20 h^{-1}. This resulted in a cycle time of 2 days per cycle. Upon addition of 1 g L^{-1} of acetic acid to the culture, the cycle time increased to 4 days and the specific growth rate decreased to 0.14 h^{-1}. During the next 3 cycles at 1 g L^{-1} acetic acid, the specific growth rate increased again to 0.17 h^{-1} and the cycle time was reduced to just over 2 days.

Subsequent increases of the acetic-acid concentration to 2, 3, 4 and 5 g L^{-1}, respectively, resulted in qualitatively similar trends: (i) upon each increase of the acetic-acid concentration, the specific growth rate in the subsequent cycle was reduced and the lag phase extended, resulting in increased cycle times; (ii) during the cycles in between the increases in the acetic-acid concentration, the specific growth rate increased and the cycle

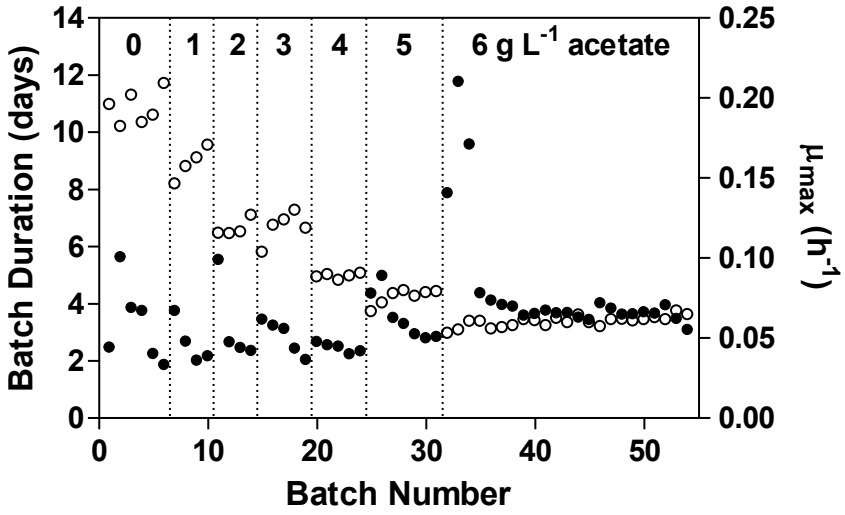


Figure 1. Selection of xylose fermenting *S. cerevisiae* strains for improved acetic-acid tolerance in anaerobic sequential batch cultivation on synthetic medium with 20 g L^{-1} xylose at increasing concentrations of acetic acid (0-6 g L^{-1}) and pH 4. An aerobic xylose grown shake-flask culture of RWB218 was used as an inoculum for the first batch fermentation. For subsequent fermentations, the cumulative CO_2 production was used to determine the automated removal ~99.5% of the culture broth and refill of the culture with fresh medium. Each point indicates the batch duration (●) and maximum specific growth rate (μ_{max}, ○) of one complete batch fermentation.

time consistently showed a downward trend. Further increasing the acetic-acid concentration to 6 g L^{-1} caused a drastic increase of the lag phase and thereby of the cycle time (Figure 1). At the end of the 54 batch fermentations, corresponding to over 400 generations based on the culture average, the cycle time had decreased again to 4 days. Compared to the initial cycles grown in the absence of acetic acid, the specific growth rate on xylose was reduced by 3-fold (0.06 h^{-1} at 6 g L^{-1} acetic acid).

Growth-regulating pH feedback in prolonged continuous cultures at increasing acetic-acid concentrations

For the second selection strategy tested in this study, *S. cerevisiae* RWB218 was cultivated in an anaerobic xylose-limited continuous culture (D=0.05 hr^{-1}) without pH control. The pH of the ingoing fresh synthetic medium was 4.25. As ammonia, the sole

nitrogen source in these cultures, is consumed, protons are released into the medium ($NH_4^+ \rightarrow NH_3 + H^+$), thereby causing a decrease in extracellular pH and a concomitant increase of the undissociated acetic-acid concentration (pK_a=4.76). As soon as the concentration of undissociated acetic acid becomes inhibitory, the specific growth rate will decrease below the dilution rate, resulting in decreased ammonium consumption and an increase of the culture pH due to dilution with fresh medium. This leads to an intrinsic growth-regulating feedback loop that provides a constant selection pressure for cells with a higher tolerance to (undissociated) acetic acid, which can continue to grow and acidify the culture broth when growth of other cells is already inhibited.

At the initial acetic-acid concentration of 1 g L^{-1}, the biomass yield on xylose was just under 0.05 g biomass (g xylose)$^{-1}$, corresponding to a specific xylose-consumption rate of 0.97 g xylose (g biomass)$^{-1}$ h^{-1} and resulting in a specific ethanol production rate of 0.36 g ethanol (g biomass) h^{-1} (Figure 2). Over the course of 8 months, representing at least 370 generations, the acetic-acid concentration in the supplied medium was periodically increased, from an initial concentration of 1 g L^{-1} to a final concentration of 5 g L^{-1} (Figure 2A). These increases initially resulted in increased biomass-specific xylose-consumption rates and reduced biomass yields on xylose, consistent with an increased ATP demand for cellular homeostasis. After 125 d, when an acetic-acid concentration of 4 g L^{-1} was reached, the xylose-consumption rate had increased by 75% from 0.97 to 1.7 g (g biomass·h)$^{-1}$, which is the highest xylose uptake flux hitherto reported for xylose-isomerase based, engineered *S. cerevisiae* (Figure 2). The ethanol production rate had increased by a similar factor from 0.36 to 0.63 g ethanol (g biomass)$^{-1}$ h^{-1} (Figure 2). A further increase of the acetic-acid concentration to 5 g L^{-1} did not result in a further increase

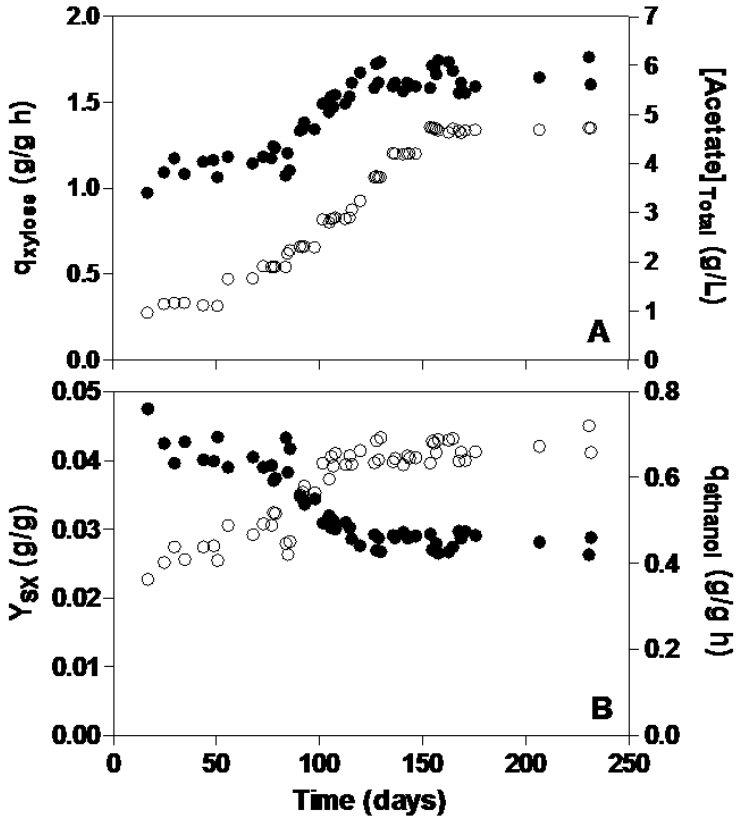


Figure 2. Selection of xylose fermenting *S. cerevisiae* strains for improved acetic-acid tolerance in anaerobic xylose-limited continuous cultivation without pH control. The physiological parameters represented are: specific xylose consumption rate (●, panel A), total acetic-acid concentration (○, panel A), the biomass yield on xylose (●, panel B) and the specific ethanol production rate (○, panel B).

of the specific xylose-consumption rate. Remarkably, this did not result in culture washout but, instead, to a steady-state culture that showed approximately the same xylose-consumption rate as observed in the cultures grown at 4 g L^{-1} of acetic acid.

Apparent instability of selected phenotypes after storage and cultivation under non-selective conditions

Culture samples taken at the end of the SBR and continuous-culture selection experiments were stored at -80 ^{o}C. Before further characterization, frozen samples were grown on xylose in aerobic shake-flask cultures without added acetic acid. Upon reaching exponential phase, these shake flasks were used to inoculate anaerobic bioreactors in which

the conditions were similar to those in the final stages of the selection experiments (20 g L^{-1} xylose, 5 g L^{-1} acetic acid, pH 4; see Materials and methods). Even after one week, neither growth nor xylose consumption were detected. This suggested that the acetic-acid tolerance acquired as a result of both selection strategies, which enabled growth on xylose at low pH at acetic-acid concentrations where such growth was not observed before, was not stable.

Acetic-acid gradient feeding demonstrates inducible acetic-acid tolerance in selected strains

The apparent loss of the acquired acetic-acid tolerance described above does not necessarily imply that tolerance is completely lost, e.g. as a result of an unstable genetic or epigenetic change. Instead, the acquired tolerance might require induction by acetic acid and thus not be expressed adequately when cells are transferred abruptly from a medium without acetic acid to a medium with a high concentration of acetic acid. To investigate the latter possibility, the parental strain *S. cerevisiae* RWB218 and aliquots from both the SBR and continuous selections runs were tested in anaerobic, xylose-limited and pH-controlled continuous cultures in which the acetic-acid concentration was increased linearly from 0 to 7 g L^{-1} over a period of 8 days (200 h). During the batch phase preceding the gradient, the specific growth rate of all three cultures was identical at 0.17 h^{-1}. Furthermore, xylose-consumption rates (0.55-0.62 g xylose (g biomass)$^{-1}$ h^{-1}) and biomass yields on xylose (0.08-0.09 g biomass (g xylose)$^{-1}$) were very similar for RWB218 and evolved cultures in xylose-limited chemostat cultures (D = 0.05 h^{-1}) without added acetic acid. Interestingly, the residual xylose concentration was much lower in chemostat cultures of the continuously evolved culture (0.47 g L^{-1}), compared to the SBR evolved culture (0.82 g L^{-1}) and especially

compared to RWB218 (1.30 g L^{-1}). This indicated that both evolution runs resulted in an improved affinity ($q_{s,max}/K_S$ [21]) for xylose, with the most pronounced improvement occurring in the culture evolved under xylose limitation.

After reaching steady state in the absence of acetic acid, the linear acetic-acid gradient was started (Figure 3). During the first three days of the acetic acid gradient, the parental strain RWB218, which was not selected for acetic-acid tolerance showed increasing xylose-consumption rates. As expected from Monod kinetics for the limiting nutrient, the residual xylose concentration increased to 5.6 g L^{-1} (Figure 3A). With less xylose available for fermentation and biomass formation, the ethanol concentration and the culture dry weight decreased. When, after three days, the acetic-acid concentration reached 2.5 g L^{-1}, the specific xylose-consumption rate of RWB218 peaked at 1.0 g xylose (g biomass)$^{-1}$ h^{-1} (Figure 3D). Subsequently, specific-xylose-consumption rates sharply decreased, reflecting the inability of this strain to deal with high acetic-acid concentrations.

As could be expected from the slightly improved affinity of the SBR-selected culture in xylose-limited chemostat cultures, its residual xylose concentration remained lower during the first three days than in the parental strain RWB218 (Figure 3B). This resulted in slightly higher ethanol concentrations. However, the specific xylose-consumption rate peaked at almost the same acetic-acid concentration (2.5 g L^{-1}) at a value of 1.1 g_{xylose} $g_{biomass}^{-1}$ h^{-1} and decreased in a pattern that was highly similar to that observed with the RWB218 strain (Figure 3D). This indicated that prolonged selection in the SBR cultures did not lead to a stable acetic-acid tolerant phenotype.

The culture selected in the continuous-culture set-up without pH control showed a completely different response to the acetic-acid gradient. During the first three days, it still responded similarly to the other strains, albeit at much lower xylose concentrations due to its improved affinity for xylose. However, where the other cultures demonstrated a sharp peak in the xylose-consumption rate, this culture reached xylose-consumption rate of just above 1.2 g xylose (g biomass)$^{-1}$ h^{-1} and maintained this flux for the next two days up to acetic-acid concentrations of 5 g L^{-1} (Figure 3D). Although the xylose-consumption rate remained constant during this period, the residual xylose concentration increased from 1.2 g L^{-1} after three days to 3.0 g L^{-1} after 5 d, indicating an impact of acetic acid on the affinity for xylose. With less xylose available for growth and metabolism, both the biomass concentration and ethanol concentration decreased during this period (Figure 3C). Although slowly decreasing, the xylose-consumption rate remained above 0.9 g_{xylose} $g_{biomass}^{-1}$ h^{-1}, until an acetic-acid concentration of 6 g L^{-1} was reached. At even higher concentrations the xylose consumption flux rapidly decreased and the culture washed out. These results demonstrate that selection in the continuous cultures without pH control resulted in a stable, acetic-acid-inducible acetic-acid tolerance.

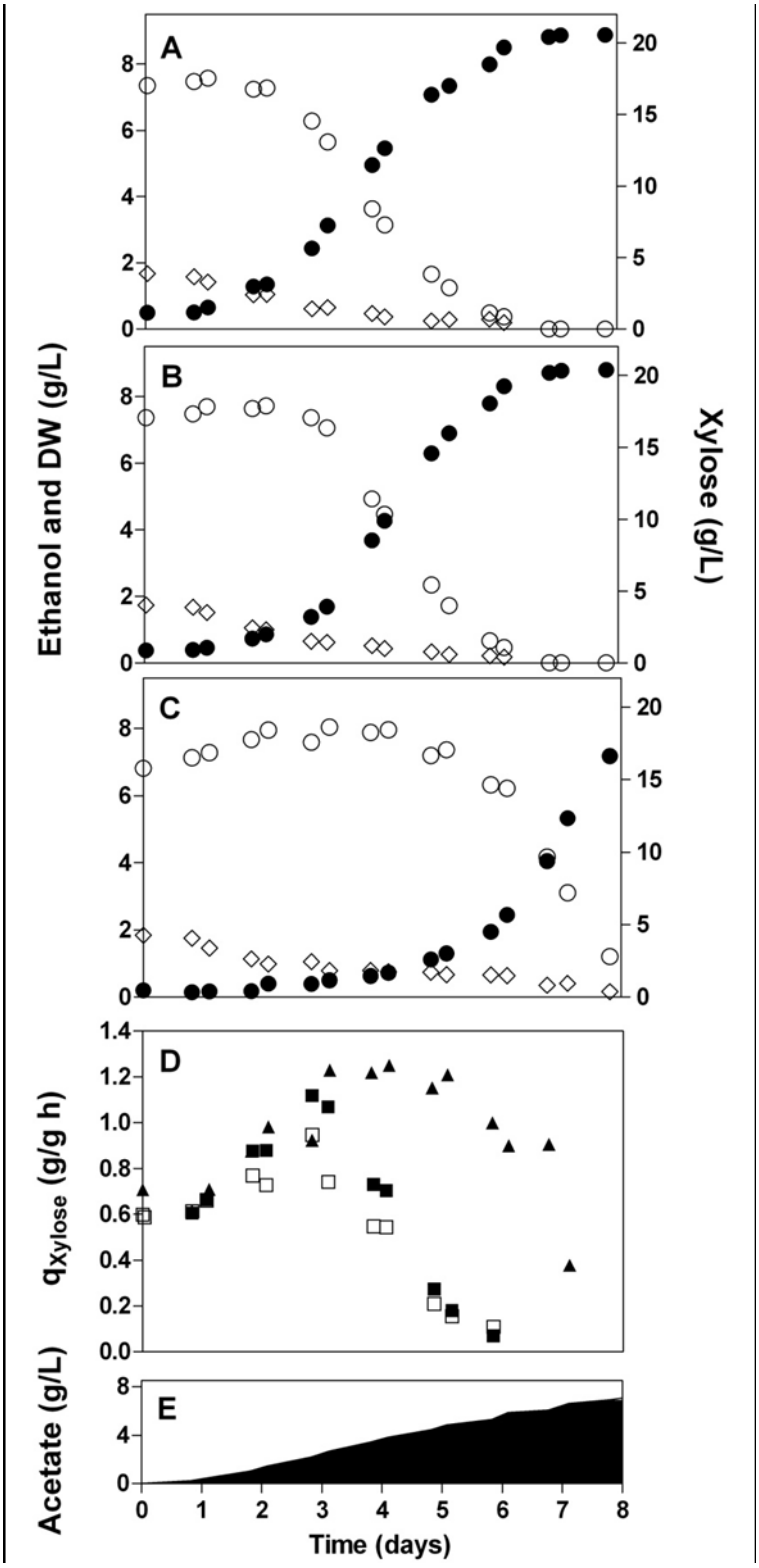


Figure 3. Impact of acetic-acid gradients in continuous cultivation of xylose-fermenting *S. cerevisiae* strain RWB218 (panel A) and two cultures selected for improved acetic-acid tolerance in either sequential batch cultivation (panel B) or continuous cultivation without pH control (panel C). Indicated are the culture dry weight (◊), the residual xylose concentration (●) and the observed ethanol concentration (○). The specific xylose-consumption rates (g xylose (g biomass)$^{-1}$ h^{-1}) are indicated in panel D for xylose-fermenting *S. cerevisiae* strain RWB218 (□), sequential batch reactor culture (■), and continuous cultivation without pH control (▲). The acetic-acid concentration increased over a period of 8 days from 0 g L^{-1} to 7 g L^{-1}.

Discussion

Prolonged cultivation of xylose-fermenting *S. cerevisiae* strains at increasing concentrations of acetic acid led to adapted cultures that grew and efficiently fermented xylose at total acetic-acid concentrations of up to 6 g L^{-1} at pH 4. These concentrations were much higher than those that allowed growth of the original xylose-fermenting strain *S. cerevisiae* RWB218. In contrast to previous reports on xylose-fermenting strains that are based on expression of heterologous xylose reductase and xylitol dehydrogenase [22] the presence of acetic acid did not result in xylitol formation by the xylose-isomerase based strain *S. cerevisiae* RWB218. This demonstrates that high-rate, high-yield ethanol production from xylose by engineered *S. cerevisiae* in the presence of high acetic-acid concentrations is intrinsically possible. This is an important conclusion for development of yeast-based processes for fermentation of lignocellulosic hydrolysates, in which acetic acid is an important inhibitory compound. Although selection in the sequential batch and continuous cultures led to a similar degree of acetic-acid tolerance, fermentation kinetics during the selection experiments revealed clear differences.

In the sequential batch cultures, each increase of the acetic-acid concentration caused an initial strong increase of the overall fermentation length. The decrease of the fermentation length in subsequent cultivation cycles was not solely due to an increase of the maximum specific growth rate but also and in particular to changes in the lag phase. Lag phases were unexpected in this cultivation system, since the automated replacement of medium was designed to maintain exponential growth. Their occurrence may be linked to the kinetics of xylose fermentation by *S. cerevisiae* RWB218. Automated medium

replacement was initiated when at least 80% of the initial xylose (20 g L^{-1}) was consumed, leaving a residual concentration below 6 g L^{-1} (0.04 mM), which is below the K_m for xylose uptake by acetate-unadapted *S. cerevisiae* RWB218 (K_m = 0.1 M, [8]). The rate of sugar fermentation is a key determinant of acetate tolerance in xylose-fermenting *S. cerevisiae* [19]. A suboptimal xylose-uptake rate towards the end of each cycle may therefore have led to an increased sensitivity to acetic acid and thus explain the observed lag phases. Consistent with the experimental data (Figure 1), this effect is expected to be most pronounced when the acetic-acid concentration is increased in a subsequent cycle.

During the other selection strategy in continuous-culture, a biphasic relation was observed between the acetic-acid concentration in the cultures and the specific rates of xylose fermentation (Figure 2C). Initially, the specific rate of xylose fermentation increased with increasing acetic-acid concentration, consistent with the key role of ATP in acetic-acid tolerance [19]. The specific rate of 1.7 g_{xylose} $g_{biomass}^{-1}$ h^{-1} reached at an acetate concentration of 4 g L^{-1} is the highest xylose fermentation rate hitherto reported for engineered *S. cerevisiae*. Surprisingly, although the cultures continued to grow at acetate concentrations above 4 g L^{-1}, the specific rate of xylose consumption did not increase further. This strongly suggests selection for 'passive' mechanisms for acetic-acid tolerance, such as a decreased permeability of the cell envelope or a decreased sensitivity of intracellular targets for acetate inhibition.

When samples from both selection experiments were stored at -80 ^{o}C and subsequently pregrown in shake flasks on xylose in the absence of acetic-acid stress, they failed to grow in anaerobic batch cultures (pH 4) supplemented with 5 g L^{-1} acetic acid. This almost complete loss of the high-level acetic-acid tolerance observed during selection is

unlikely to be due to reversion of mutations in view of the limited number of generations (± 10) of non-selective growth. There is a rapidly growing evidence for the occurrence of bi- or multistable situations, even in genetically homogeneous cultures [23]. Such multistability, which can be a direct consequence of the architecture of regulatory or catalytic networks, may be responsible for the rapid reversion to acetate sensitivity upon a change in growth conditions.

In contrast, when similarly pregrown samples were subjected to a linearly increasing acetic acid-concentration a drastically increased acetic-acid tolerance was observed for the evolved continuous culture (Figure 3C). Apparently, selection in the continuous cultures resulted in (hyper)inducible rather than constitutive acetic-acid tolerance. Acetic acid occurs in natural environments of *S. cerevisiae* and is itself a product of anaerobic yeast metabolism. Indeed, *S. cerevisiae* is known to express inducible tolerance mechanisms, such as those induced by acetate-induced *HAA1* regulon [24, 25]. The inducible acetate tolerance of the continuous-culture selected cells may therefore, for example, have resulted from an increased copy number of such acetate-inducible tolerance genes. Interestingly, evolutionary engineering of *S. cerevisiae* for tolerance to furfural, another inhibitor of yeast metabolism that is formed during lignocellulose hydrolysis, yielded a furfural-tolerant phenotype that was retained during cultivation in the absence of furfural [26]. Since furfural is formed under non-physiological physicochemical conditions, yeast is unlikely to have evolved specific furfural-inducible resistance mechanisms and evolved resistant phenotypes are more likely to be based on constitutively expressed mutations.

The inducible acetic-acid tolerance obtained in the continuous selection system is impractical from an applied point of view, since incorporation of an acetic-acid adaptation

step into industrial ethanol production processes represents an undesirable complication. However, strains with inducible tolerance, obtained via the continuous selection procedure described in this study, provide an interesting starting point to develop strains with constitutive acetate tolerance, either via classical strain improvement (e.g. mutagenesis and selection) or via reverse engineering of acetic-acid tolerance after analysis of the molecular basis of their inducible tolerance by genome-wide analysis techniques.

Acknowledgements

We thank Vincent de Vrind for his contributions during strain characterization. AW acknowledges support from Swiss National Science Foundation, the Swiss Initiative in Systems Biology (SystemsX), and the Santa Fe Institute.

References

1. U. Sauer. Evolutionary engineering of industrially important microbial phenotypes. *Metabolic Engineering*, pages 129–169, 2001.
2. M. Valli, M. Sauer, P. Branduardi, N. Borth, D. Porro, and D. Mattanovich. Improvement of lactic acid production in *Saccharomyces cerevisiae* by cell sorting for high intracellular pH. *Applied Environmental Microbiology*, 72(8):5492–5499, 2006.
3. A. Novick and L. Szilard. Description of the chemostat. *Science*, 112(2920):715–716, 1950.
4. P. Daran-Lapujade, J-M. Daran, A.J.A. van Maris, J.H. de Winde, and J.T. Pronk. Chemostat-based micro-array analysis in baker's yeast. *Advances in Microbial Physiology*, 54:257–311, 414–417, 2008.
5. H.W. Wisselink, M.J. Toirkens, Q. Wu, J.T. Pronk, and A.J.A. van Maris. Novel evolutionary engineering approach for accelerated utilization of glucose, xylose, and arabinose mixtures by engineered *Saccharomyces cerevisiae* strains. *Applied Environmental Microbiology*, 75(4):907–914, 2009.
6. J.R.M. Almeida and B. Hahn-Hägerdal. Developing *Saccharomyces cerevisiae* strains for second generation bioethanol: improving xylose fermentation and inhibitor tolerance. *International Sugar Journal*, 111(1323):172–180, 2009.
7. M. Sonderegger and U. Sauer. Evolutionary engineering of *Saccharomyces cerevisiae* for anaerobic growth on xylose. *Applied Environmental Microbiology*, 69(4):1990–1998, 2003.
8. M. Kuyper, M.J. Toirkens, J.A. Diderich, A.A. Winkler, J.P. Dijken, and J.T. Pronk. Evolutionary engineering of mixed-sugar utilization by a xylose-fermenting *Saccharomyces cerevisiae* strain. *FEMS Yeast Research*, 5(10):925–934, 2005.
9. H.B. Klinke, A.B. Thomsen, and B.K. Ahring. Inhibition of ethanol-producing yeast and bacteria by degradation products produced during pre-treatment of biomass. *Applied Microbiology and Biotechnology*, 66(1):10–26, 2004.
10. E. Palmqvist, H. Grage, N.Q. Meinander, and B. Hahn-Hägerdal. Main and interaction effects of acetic acid, furfural, and p-hydroxybenzoic acid on growth and ethanol productivity of yeasts. *Biotechnology and Bioengineering*, 63(1):46–55, 1999.
11. E. Palmqvist and B. Hahn-Hägerdal. Fermentation of lignocellulosic hydrolysates. II: inhibitors and mechanisms of inhibition. *Bioresource Technology*, 74(1):25–33, 2000.
12. J. Zaldivar, J. Nielsen, and L. Olsson. Fuel ethanol production from lignocellulose: a challenge for metabolic engineering and process integration. *Applied Microbiology and Biotechnology*, 56(1):17–34, 2001.
13. L.H.A. Lima, M. das Graças de Almeida Felipe, M. Vitolo, and F.A.G. Torres. Effect of acetic acid present in bagasse hydrolysate on the activities of xylose reductase and xylitol dehydrogenase in *Candida guilliermondii*. *Applied Microbiology and Biotechnology*, 65(6):734–738, 2004.
14. M. Mollapour and P.W. Piper. Hog1 mitogen-activated protein kinase phosphorylation targets the yeast Fps1 aquaglyceroporin for endocytosis, thereby rendering cells resistant to acetic acid. *Molecular and Cellular Biology*, 27(18):6446–6456, 2007.
15. M. Mollapour, A. Shepherd, and P.W. Piper. Novel stress responses facilitate *Saccharomyces cerevisiae* growth in the presence of the monocarboxylate preservatives. *Yeast*, 25(3):169–177, 2008.
16. M.E. Pampulha and M.C. Loureiro-Dias. Activity of glycolytic enzymes of *Saccharomyces cerevisiae* in the presence of acetic acid. *Applied Microbiology and Biotechnology*, 34(3):375–380, 1990.

17. P. Piper, C.O. Calderon, K. Hatzixanthis, and M. Mollapour. Weak acid adaptation: the stress response that confers yeasts with resistance to organic acid food preservatives. *Microbiology*, 147(10):2635–2642, 2001.

18. K.C. Thomas, S.H. Hynes, and W.M. Ingledew. Influence of medium buffering capacity on inhibition of *Saccharomyces cerevisiae* growth by acetic and lactic acids. *Applied Environmental Microbiology*, 68(4):1616–1623, 2002.

19. E. Bellissimi, J.P. van Dijken, J.T. Pronk, and A.J.A. van Maris. Effects of acetic acid on the kinetics of xylose fermentation by an engineered, xylose-isomerase-based *Saccharomyces cerevisiae* strain. *FEMS Yeast Research*, 9(3):358–364, 2009.

20. C. Verduyn, E. Postma, W.A. Scheffers, and J.P. van Dijken. Effect of benzoic acid on metabolic fluxes in yeasts: a continuous-culture study on the regulation of respiration and alcoholic fermentation. *Yeast*, 8(7):501–517, 1992.

21. D.K. Button. Nutrient-limited microbial growth kinetics: overview and recent advances. *Antonie van Leeuwenhoek*, 63(3):225–235, 1993.

22. S.S. Helle, A. Murray, J. Lam, D.R. Cameron, and S.J.B. Duff. Xylose fermentation by genetically modified *Saccharomyces cerevisiae* 259ST in spent sulfite liquor. *Bioresource Technology*, 92(2):163–171, 2004.

23. J.W. Veening, W.K. Smits, and O.P. Kuipers. Bistability, epigenetics, and bet-hedging in bacteria. *Annual Review of Microbiology*, 62(1):193–210, 2008.

24. D.A. Abbott, E. Suir, A.J.A. van Maris, and J.T. Pronk. Physiological and transcriptional responses to high concentrations of lactic acid in anaerobic chemostat cultures of *Saccharomyces cerevisiae*. *Applied Environmental Microbiology*, 74(18):5759–5768, 2008.

25. A.R. Fernandes, N.P. Mira, R.C. Vargas, I. Canelhas, and I. Sá-Correia. *Saccharomyces cerevisiae* adaptation to weak acids involves the transcription factor Haa1p and Haa1p-regulated genes. *Biochemical and Biophysical Research Communications*, 337(1):95–103, 2005.

26. D. Heer, D. Heine, and U. Sauer. Resistance of *Saccharomyces cerevisiae* to high concentrations of furfural is based on NADPH-dependent reduction by at least two oxireductases. *Applied Environmental Microbiology*, 75(24):7631–7638, 2009.